HIGHWAY INFORMATION SYSTEM

RELEASE 4.Ø

USER'S MANUAL

Department of Civil Engineering & Engineering Mechanics Montana State University Bozeman

HIGHWAY INFORMATION SYSTEM

RELEASE 4.Ø

USER'S MANUAL

Prepared for the:

STATE OF MONTANA
DEPARTMENT OF HIGHWAYS
PLANNING AND RESEARCH BUREAU

In cooperation with the:

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL HIGHWAY ADMINISTRATION

Project Director: Ralph W. Zimmer, P.E.

Prepared by:

Larry J. Coats, Edward G. Knoyle, and Ralph W. Zimmer

DEPARTMENT OF CIVIL ENGINEERING AND ENGINEERING MECHANICS
MONTANA STATE UNIVERSITY
Bozeman, Montana 59715

April, 1976

FOREWORD

This report is a portion of the documentation of Release 4.Ø of the Highway Information System undertaken by the Department of Civil Engineering and Engineering Mechanics, Montana State University. The retrieval system has been evolving over the last several years under the sponsorship of the Planning and Research Bureau of the Montana Department of Highways with some assistance from the Highway Traffic Safety Division, Montana Department of Community Affairs.

Release 4.Ø of the Highway Information System is documented in the following volumes:

Highway Information System Release 4.Ø: System Overview

Provides an introduction to the Highway Information System.

Highway Information System Release 4.Ø: Index

Provides an index to all manuals except the System Overview and Program Listings.

Highway Information System Release 4.Ø: User's Manual

Describes how to use the Highway Information System for retrieving information and for printing reports and summaries.

Highway Information System Release 4.Ø: Data Coding Manual

Describes the data card formats for entering data into the Highway Information System files.

Highway Information System Release 4.Ø: System Maintenance Manual

Provides information for performing scheduled system backups and file reorganizations and for allocating system files.

Highway Information System Release 4.Ø: Record Formats & Subroutines

Describes the internal record formats of the various files and provides calling sequences to subroutines. This manual is intended for persons writing new programs to add to the Highway Information System.

Highway Information System Release 4.Ø: Programming Details

Describes the existing programs and provides a guide to the program listings. This manual is intended for persons maintaining existing software in the Highway Information System.

Highway Information System Release 4.Ø: Program Listings

Contains computer-generated listings of all source programs of the Highway Information System.

Although the project was conceived, initiated, and primarily funded through the Planning and Research Bureau of the Montana Department of Highways, the development cost of selected portions of the system was borne by the Highway Traffic Safety Division of the Montana Department of Community Affairs.

In developing the system, the CE & EM Department has had the privilege of using an IBM OS/VS1 37Ø/145 computer located at the Data Processing Bureau of the Montana Department of Highways in Helena. PL/I has been used for most of the programs because of its versatility and ease of use. BAL (assembler) has been used for most input-output modules and for other modules that require its increased capabilities and efficiency over PL/I.

The project could never have progressed to its current state without the continued and patient encouragement and assistance from the Planning and Research Bureau and the Data Processing Bureau of the Montana Department of Highways, and from the Highway Traffic Safety Division of the Department of Community Affairs.

The project conclusion was also hastened by the significant effort of other project personnel: Scott H. Danforth, R. Helene Knowlton, and Doug M. Geiger.

T A B L E   O F   C O N T E N T S

LIST OF CHAPTERS

TABLE OF CONTENTS

T A B L E   O F   C O N T E N T S

LIST OF FIGURES

LIST OF FIGURES (continued)

T A B L E   O F   C O N T E N T S

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This publication describes the use of release 4.Ø of the Highway Information System (HIS). It is intended for all potential users of HIS, regardless of technical background.

The publication Highway Information System Release 4.Ø - System Overview is an overall introduction to HIS. This publication should be read before attempting to utilize any other HIS publications.

Release 4.Ø of HIS is a batch-processing and remote batch-processing system with no on-line support. HIS has been designed for compatibility with future on-line support and a minimum of conversion effort will be needed to provide on-line functions when computer hardware and software become available.

The Highway Information System is composed of a set of data files and a set of computer programs. Computer programs are provided for the following functions:

1. Printing summaries and listings from the data files.
2. Updating and maintaining the data files.
3. Storing backup copies of the data files and restoring the data files from backup copies.

This manual describes the programs that print summaries and listings. The update and maintenance software is described in the publication Highway Information System Release 4.Ø - Data Coding. The backup procedures are described in the publication Highway Information System Release 4.Ø - System Maintenance.

The HIS files are designed to eliminate data redundancy by storing each data item in one and only one file. This organization helps promote data integrity, helps force file compatibility, and helps simplify file maintenance. The files are organized around a common reference method to allow cross-referencing of two or more files when producing reports and summaries.

A great deal of effort has been expended orienting HIS toward the user rather than toward the computer programmer. The user prepares a "command" that indicates his needs and submits the command to the computer along with a minimal amount of IBM Job Control statements. The system can be used by persons unfamiliar with computers or with the internal workings of HIS.

The HIS Files

HIS contains a number of individual files. The files utilize a common reference method so that several files can be accessed simultaneously to derive information for summaries and reports.

The Roadlog File - The roadlog file contains physical roadway information pertaining to all federal aid and local system routes. The file is organized around the common key of HIS - the route system, route number and milepoint. Each record contains the key of a roadlog "section break" (any location at which the physical characteristics of the roadway change, any jurisdictional break such as a county line or city limits, any major junction, etc.). The record describes one section of highway. When a key is provided to the system, the roadlog record containing data for that key can be directly retrieved. When two routes are coincident, a cross-reference technique is used to prevent duplication of data.

The True Mileage File - The true mileage file is the basic tie between reference posts and actual roadway locations. The file provides the location of each reference post with respect to the beginning of the route on which it is located.

The Traffic File - The traffic file contains "average annual daily traffic" (AADT) counts at each count station for three separate years. Each count station is identified by its route system, route number, and milepoint.

The Accident Files - Each reported accident that occurs in the state is entered into the accident files. All legally reportable accidents that are reported are entered. Many accidents that are not legally reportable are also entered. The data includes both accidents that were investigated by law enforcement officers and those that were not. The accident files are organized around an assigned accident number rather than the standard route system, route number, and milepoint to insure a unique key and because not all accidents occur on routes with reference posts. A directory file is maintained that allows access by route system, route number, and milepoint.

The Sufficiency File - The sufficiency file contains data elements that indicate quality and condition of roadway sections. The records are keyed by route system, route number, and milepoint.

The Bridge File - The bridge file contains an entry for each bridge in the state. Data elements within the entry describe the type and condition of the bridge. The bridge file is keyed on route system, route number, and milepoint.

The Railroad Crossing File - The railroad crossing file contains an entry for each railroad crossing in the state. Each entry describes the physical characteristics of a railroad crossing. The crossings are keyed by route system, route number, and milepoint.

The Skid File - The skid file contains the results of skid trailer tests on the state's highways. Rural tests and many municipal on-system tests are keyed on route system, route number, and milepoint. The remaining municipal tests are keyed on city coordinates to allow future correlation of data with the accident files.

The Urban Sign Inventory Files - These files contain sign inventory data for cities participating in the urban sign inventory system. Each city has a separate file containing its own data. The files are keyed on an assigned sign assembly number rather than on location data. Currently these files are not completely compatible with other HIS files but it is expected that the data will be able to correlate with accident data at some future date.

Other "Base" Files - Several small files are maintained in the system for uses as cross-references, such as the grid table and the defense section cross-reference. These files are described in chapter 11 of this manual. The Data Processing Bureau has implemented several files into the system, including a rural sign inventory file, a summons file, and an open-range file.

Other Files - Several files are constructed on an as-needed basis from the base files. These include such files as the accident-by-sections file and the traffic-by-sections file. Information on the construction of these files is included in the publication Highway Information System Release 4.Ø - System Maintenance.

Introduction to the Use of HIS

HIS is a user-oriented system which can be used by persons having little or no technical background with computers. The user prepares a command following standard formats described in this manual and submits it to the computer with a minimal amount of IBM Job Control Language (JCL).

Job Control Language - An Operating System (sometimes called a Supervisor) is always supplied by the vendor for large computer systems. IBM provides a number of different Operating Systems so that each installation can use one that fits its needs. HIS is currently designed to run under the OS/VS1 operating system, but can be readily adapted to OS/VS2, OS/MFT, or OS/MVT with only minor changes. This section on Job Control Language (JCL) is applicable to OS/VS1, but the JCL of the other three systems are very similar.

Each batch-submitted user request to the operating system is called a "job," and can consist of one or more "job steps." The first control card of each job is a "JOB" card. The JOB card contains accounting information and certain other control information. Before you can submit any runs to the computer you must obtain a JOB card from the Data Processing Bureau.

Each job step consists of an "EXEC" card plus zero or more "DD" cards. The EXEC card identifies to the operating system what program(s) are to be run. The DD cards identify to the operating system each of the data files that are accessed by the programs. Fortunately, most of the complicated JCL can be stored within a computer file so that very little JCL is needed in a user-oriented system. The standard HIS job setup consists of the following control cards:

```
// JOB (a JOB card is obtained from the Data Processing Bureau)
// EXEC catproc
//SYSIN DD *

  One or more HIS commands

/*
```

The JOB card contains the accounting information. The EXEC statement indicates to the Operating System that the Highway Information System is needed. The SYSIN DD statement indicates the beginning of the user's commands. The '/*' card indicates the end of the user's commands. "catproc" on the EXEC statement is replaced by the name of one of the HIS JCL files:

HISRLG	Roadlog subsystem
HISTRAF	Traffic and true mileage subsystems
HISACCA	Accident subsystem - rural accident analysis
HISACCM	Accident subsystem - municipal accident analysis
HISACC	Accident subsystem - other reports
HISBRID	Bridge subsystem
HISRRX	Railroad subsystem
HISSKID	Skid subsystem
HISSUFF	Sufficiency subsystem
HISGRID	Grid table subsystem
HISUSN	Urban sign subsystem
HIS	Miscellaneous programs

Each of these names is the name of a "cataloged procedure" (a file containing job control information). Listings of the cataloged procedures are included in Appendix A.

HIS Commands - Commands are used to describe to HIS what outputs are needed. The basic command format is:

```
:prognam,keyword=option,keyword=option,...
```

The colon in column 1 indicates a HIS command. "prognam" is the name of one of the HIS programs. After the program name is zero or more "parameters" coded in the format "keyword=option". A sample command is:

```
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
```

This command requests an output by the LIST program. It requests a listing of the roadlog data stored for interstate route number 9Ø.

The command begins in column 1 with a colon and continues with no imbedded blanks until it ends. The only exception is when continuation cards are used. After any comma that separates two parameters or that separates the program name from the first parameter, leave the remainder of the card blank. Begin a second card with a colon and one or more blanks. Then continue the command. For example, the above command can be written as:

```
:LIST,
:  FILE=ROADLOG,
:  DATA=INTERSTATE=9Ø
```

The command has to be combined with JCL to submit to the computer. For example, the complete job setup for the above LIST command is:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
/*
```

This manual contains a sample job setup for each of the available summary and report programs.

Comment cards may be included between commands, before the first command, or after the last command. Code a "greater-than" sign (>) in column 1. The remainder of the card can contain any desired characters. The comment cards are printed with the command listing but are otherwise ignored.

<u>Print Options</u>

Users of the Highway Information System have considerable control over the format of printed output. Formatting options are specified in either of two manners: (1) parameters coded on the EXEC card, and (2) parameters coded on the command card. In general, parameters coded on the EXEC card are in effect for all commands submitted in a run while parameters coded on a command card are in effect only for that command.

Output to a CRJE Terminal - If you are submitting a run from a CRJE terminal, you can receive your output at your terminal, from the computer's printer, or at both places. To receive your output at the CRJE terminal only, code PARM='PRINTER-DD=CRJE' on the EXEC statement. To receive your output at both the computer's printer and at your terminal, code PARM='OUTPUT-FILE=CRJE' on the EXEC statement. If you do not code one of these parameters, the output will be printed on the computer's printer only. An example of a job setup to route the output to both places is:

```
// JOB
// EXEC HISRLG,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
/*
```

Output to Special Forms - The PRINTER-DD and/or OUTPUT-FILE parameters can be used to route output to special forms. If you use PRINTER-DD, the only output generated is on the special forms. If you use OUTPUT-FILE, the output is written both to the special form and to the standard form. The following standard names are implemented:

WHITE1 - White 11x14 one-part form

GREEN3 - Green-striped 11x14 three-part form

A sample job setup is:

```
// JOB
// EXEC HISRLG,PARM='PRINTER-DD=WHITE1'
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
/*
```

To use forms other than these, you must be familiar with the conventional form numbers in use by the Data Processing Bureau - a list of these can be obtained through the Data Processing Bureau. You can then override the standard form by using the following job setup:

```
// JOB
// EXEC HISRLG,FORM='(class,,form-number)'
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
/*
```

Obtaining More than One Copy - Two methods are available for obtaining multiple copies, and they can be used together if desired. The first method is to request a special form consisting of 2-, 3-, or 4-part paper. A list of forms available can be obtained from the Data Processing Bureau and can be requested in your run as shown above. The second method is to specify COPIES=n on the EXEC card. The printed output is written to the printer n times. An example of a job setup using COPIES is:

```
// JOB
// EXEC HISRLG,COPIES=4
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
/*
```

Storing Output on Tape or Disk - If you are familiar with JCL, you can store output on tape or disk to be copied to the printer later. Use the PRINTER-DD parameter if you wish only the file, or the OUTPUT-FILE parameter if you wish both one printed copy and the file. Specify the name of the DD statement you are providing for the file in the parameter. An example job setup is:

```
// JOB
// EXEC HISRLG,PARM='OUTPUT-FILE=SAVE'
//SAVE DD ...
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø
/*
```

The file can be copied to the printer at a later time with the following job setup:

```
// JOB
// EXEC HISX133
//SYSIN DD ...
```

When using HISX133, you can specify FORM='(class,,form-number)' on the EXEC statement as shown above in "Output to Special Forms." You can obtain multiple copies by coding PARM=n on the EXEC statement.

Number of Lines Per Page - The standard line count is 6Ø lines per page. This value can be altered by coding PAGESIZE=n on a command, and the altered value is in effect only for the command on which it is coded. The line count can be altered for a series of commands by including a command that specifies the SYS-PARAM program as in the following job setup:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:SYS-PARAM,PAGESIZE=46
:LIST,...
:LIST,...
/*
```

The line count will be 46 for both commands.

Page Numbers - Page numbers can be printed. Once started, page numbering continues onto succeeding commands. Begin page numbering by coding PAGE-NUMBER=n on a command (this can be a SYS-PARAM command). Once begun, page numbering continues from one command to the next until a command that specifies PAGE-NUMBER=STOP is encountered. A third form of the parameter is PAGE-NUMBER=$+n, which indicates that the page number is to be incremented by "n". Consider the following job setup:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:LIST,...
:LIST,...,PAGE-NUMBER=1Ø
:LIST,...,PAGE-NUMBER=$+2Ø
:LIST,...,PAGE-NUMBER=18
/*
```

No page numbers are printed for the first command. The second command begins with page number 1Ø. A gap of 2Ø pages is left between the output of the second command and the third command. The last command begins with page number 18.

Normally, of course, page numbers are incremented by 1. You can specify an increment ranging from 1 through 9 by coding PAGE-INCREMENT=n on the command. Page numbers are printed at the top of the page, and are normally printed right-justified in columns 12Ø-124 of the printer. The column numbers of either even or odd page numbers (or both) may be specified by coding EVEN-PAGE-POSITION=n or ODD-PAGE-POSITION=n. "n" is the column number of the leftmost position of a 5-character field, and may range from 1 to 128.

Top Margins - A top margin of "n" blank lines can be printed on each page printed by a command. Code TOP-MARGIN=n on the command. To generate the same top margin on two or more commands, code TOP-MARGIN=n on a command that specifies the SYS-PARAM program.

Table Numbers - The user may request a heading in the form "TABLE NUMBER n" printed at the top of each page printed by a command. Code TABLE-NUMBER=n on the command. Table numbering, once begin, is continued onto successive commands with n incremented by one for each new command. To stop table numbering, code TABLE-NUMBER=STOP on a command.

Page Ejection - Normally, HIS ejects to a new page for each new command. The parameter PAGE-EJECT=SUPPRESS coded on a command suppresses page ejection for that command. When coded on the first command, that command's output begins on the same page as the command listing.

User Titles - You may include up to five 8Ø-character lines to be printed on each page printed by a command. Choose a name such as TITLES and code this name in a TITLE-DD parameter on the command. Enter the titles as shown in the following example:

```
// JOB
// EXEC HISRLG
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=INTERSTATE=9Ø,TITLE-DD=TITLES
/*
//TITLES DD *
  One to five title cards to be printed on each page
/*
```

A set of titles can be placed in effect for two or more successive commands by coding TITLE-DD on a command that specifies the SYS-PARAM program.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters

The DATA parameter is widely used throughout the Highway Information System. It is used for specifying the route(s) to be processed in a run. It is a flexible means for specifying one or more route systems, one or more routes within a route system, or a portion of a route. Use the following format to specify one entire route system:

$$\text{DATA=} \left\{ \begin{array}{l} \text{INTERSTATE} \\ \text{PRIMARY} \\ \text{SECONDARY} \\ \text{URBAN} \\ \text{LOCAL} \\ \text{LOCAL-MUNIC} \end{array} \right\}$$

These options specify one of the following route systems:

1. Federal aid interstate system.
2. Federal aid primary system.
3. Federal aid secondary system.
4. Federal aid urban system.
5. Local system (county roads).
6. Local municipal system (city streets).

The following formats are available for specifying multiple systems:

$$\text{DATA=} \left\{ \begin{array}{l} \text{INTERSTATE+PRIMARY} \\ \text{FEDERAL-AID} \\ \text{ALL} \end{array} \right\}$$

To request a range of routes within a route system, use one of the system names shown above and add the characters "=n-m", where n and m are the desired route numbers. Leading zeroes do not need to be coded. Examples are DATA=INTERSTATE-9Ø-94 and DATA=PRIMARY=6-8. If only one route is needed, m can be omitted as in DATA=SECONDARY=2Ø9. To process only a portion of a route, use the DATA parameter to identify the route number and use START-MILEPOINT and/or END-MILEPOINT to specify the beginning and ending milepoints. Code milepoints in the format 'nnn+n.nnn'. An example is DATA=PRIMARY=8,START-MILEPOINT=Ø1Ø+Ø.42Ø,END-MILEPOINT=Ø43+Ø.773.

CHAPTER 2

THE ACCIDENT SUBSYSTEM

Introduction

Accident data is stored in the following files:

1. The accident detail file - Contains one entry for each accident with information pertaining to the accident as a whole, such as the date occurred.
2. The accident vehicle file - Contains one or more entries for each vehicle and for each pedestrian involved in an accident with information that pertains to individual vehicles and pedestrians.
3. The accident directory file - Contains a cross-reference index that allows access of accident data by route system, route number, and milepoint.

The accident detail file contains the following data elements:

Accident number - provides a unique identifier for each accident
Date and time occurred
Date and time investigating agency notified
Date and time investigating officer arrived
City number (if accident occurred within a city)
County number
Location (route system, route number, and milepoint for rural on-system accidents, route system, route number and coordinates for municipal on-system accidents, coordinates for municipal off-system accidents, and range-township-section for rural off-system accidents)
First harmful event
First object struck off roadway
Injury severity
Damage severity
Class of trafficway
Roadway-related location (on or off roadway)
Junction-related location
Number of vehicles
Number of pedestrians
Number of fatalities
Number of injuries
Weather condition
Road condition
Light condition
Traffic controls
Other damage - type, severity, and owner
Posted speed limit
Engineering study requested
Analysis (contributing circumstances)
Type of collision
Legally reportable or non-reportable
Investigated or uninvestigated

The accident vehicle file contains the following data elements:

- Accident number - provides a link to the accident detail file
- Vehicle/pedestrian number
- Name of driver/pedestrian
- Driver license number
- Driver license state
- Date of birth
- Re-examination code
- Charge code
- Summons number
- Contributing circumstances - Vision, physical defects, road defects, mechanical defects, and possible violations
- Information on driver, pedestrian, and passengers - Age, alcohol, sex, and injury severity
- Vehicle year
- Intent
- Body and trailer style
- Vehicle involved in interstate traffic
- Vehicle license number or VIN
- Damage level (over or under $250)
- Damage severity

The accident directory file contains the following data elements:

- Route system, route number, and milepoint
- Accident number
- Number of fatalities
- Number of injuries
- Date and hour occurred
- First harmful event
- Collision type
- Road condition
- Number of lanes

Because the Highway Information System is an integrated file system, other HIS files can be accessed when processing accident data. In particular, some of the accident software accesses the roadlog, traffic, and true mileage files when performing accident analysis of rural on-system accidents. The bridge, railroad, and skid files are compatible with the accident files, but no major use of these files in accident analysis has been attempted to date.

Standard User Input - Accident Subsystem

Because the Highway Information System has been designed with the user-machine interface in mind, user input has been standardized to as great a degree as possible. To run most of the accident subsystem programs, the following job setup is submitted:

```
// JOB
// EXEC catproc
//SYSIN DD *

   One or more commands

/*
```

A job card must be assigned to you by the Data Processing Bureau, and contains accounting information needed by the computer. "catproc" is one of the following names:

HISACC - Used for most accident subsystem programs.
HISACCA - Used for rural on-system accident analysis programs.
HISACCM - Used for municipal accident analysis programs.

The job setup can be either punched onto cards and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, it is possible to direct the output to the terminal. A sample job setup is:

```
// JOB
// EXEC HISACC,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *
   One or more commands
/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To get the output to your terminal only, specify PRINTER-DD=CRJE instead of OUTPUT-FILE=CRJE.

A number of formatting options - page numbering, obtaining more than one copy, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Accident Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. The simplest command has the format ':program'. The colon identifies an input card as a HIS command card, and the "program" name coded identifies one of the programs described in this manual. For example, the command

```
:COUNT-ACCIDENTS
```

requests that the COUNT-ACCIDENTS program be run. Most commands also specify certain run-time options recognized by the specified program. These run-time options are specified by coding one or more "parameters" on the command. Each parameter has the format "keyword=option". An example is the following command:

```
:COUNT-ACCIDENTS,START-DATE=Ø1/Ø1/76,END-DATE=Ø1/31/76
```

No blanks can be imbedded in the command, except when a command is continued onto a second line. To continue a command onto another line, code a comma after a complete parameter or after the program name and leave the remainder of the line blank On the continuation card code a colon followed by one or more blanks, and then continue coding parameters. For example, the above command can be continued on three lines instead of only on one:

```
:COUNT-ACCIDENTS,
:   START-DATE=Ø1/Ø1/76,
:   END-DATE=Ø1/31/76
```

Parameters may be coded in any order on the commands.

The remainder of this section describes certain parameters that can be used on most of the commands described in this chapter.

The START-DATE and END-DATE Parameters - These parameters are optional on many accident commands and are used to select accidents by date. If neither parameter is specified no date selection is performed, and all accidents are included. When START-DATE is specified, only accidents that

occurred on or following that date are considered. When END-DATE is specified, only accidents that occurred on or before that date are considered. Note that, when both parameters are coded, the date coded in START-DATE must be equal to or earlier than the date coded in END-DATE. Each date is coded as a two-digit month, day, and year separated by slashes. For example, code November 2, 1976 as 11/Ø2/76. Leading zeroes must be coded.

<u>The START-ACCIDENT and END-ACCIDENT Parameters</u> - These parameters are used to select accidents based on accident number. If neither parameter is specified, no selection based on accident number is performed. If START-ACCIDENT is specified, only accidents having a number equal to or greater than than the specified number are included. If END-ACCIDENT is specified, only accidents having a number equal to or less than the specified number are included. Note that if both parameters are specified, the number in START-ACCIDENT must be equal to or smaller than the number in END-ACCIDENT. Code the accident number as a 12-digit number. The format of the accident number is:

(Investigated accidents)

1-2	Year occurred
3-5	Agency - ØØØ for highway patrol City number for city police County number for county law enforcement
6-8	Badge number of investigating officer
9-1Ø	Month occurred
11-12	Sequence number to ensure unique number

(Uninvestigated accidents)

1-2	Year occurred
3	R if rural accident, M if municipal accident
4-6	County number if rural accident, city number if municipal accident
7-8	Month occurred
9-12	Sequence number (one-character alphabetic and 3-digit numeric)

START-ACCIDENT and END-ACCIDENT provide an effective means of selecting accidents based on reporting agency or on a particular officer during a single year. For example, all accidents coded by Great Falls patrolmen

during the year 1976 begin with the five digits 76Ø52. The parameters START-ACCIDENT=76Ø52ØØØØØØ,END-ACCIDENT=76Ø529999999 cause only those 1976 accidents investigated by Great Falls patrolmen to be included. Similarly, the parameters START-ACCIDENT=76Ø52666ØØØØ,END-ACCIDENT=76Ø526669999 cause only 1976 accidents investigated by Great Falls patrolman number 666 to be included.

The LOCATION, CITY, and COUNTY Parameters - These three parameters provide selection based on area. Only one of these parameters can be coded on a given command. The LOCATION parameter has two formats:

$$\text{LOCATION=} \begin{Bmatrix} \text{STATEWIDE} \\ \underline{\text{EVERYTHING}} \end{Bmatrix}$$

LOCATION=EVERYTHING is the default chosen if none of the parameters LOCATION, CITY, or COUNTY are specified on a command. It specifies that no location selection is performed. LOCATION=STATEWIDE also specifies that no location selection is performed, but in addition specifies that only legally reportable accidents are desired.

The CITY parameter is used to select only those accidents occurring with the city limits of one of the incorporated cities. Code the parameter as:

CITY=city-name

where "city-name" is one of the names shown in Table 2-1. Only one city can be specified in the CITY parameter. To select accidents that occurred within any of several cities use the SELECT-DD parameter.

The COUNTY parameter is used to select only those accidents that occurred within a given county. Code the parameter as:

COUNTY=county-name

where "county-name" is one of the names shown in Table 2-2. Only one county can be specified. To select accidents that occurred within any of several counties use the SELECT-DD parameter.

TABLE 2-1

NAMES CODED IN CITY PARAMETER

City Number	Name as Coded	City Number	Name as Coded	City Number	Name as Coded
ØØ1	ALBERTON	Ø43	FLAXVILLE	Ø85	OPHEIM
ØØ2	ANACONDA	Ø44	FORSYTH	Ø86	OUTLOOK
ØØ3	BAINVILLE	Ø45	FORT-BENTON	Ø87	PHILIPSBURG
ØØ4	BAKER	Ø46	FROID	Ø88	PLAINS
ØØ5	BEARCREEK	Ø47	FROMBERG	Ø89	PLENTYWOOD
ØØ6	BELGRADE	Ø48	GERALDINE	Ø9Ø	PLEVNA
ØØ7	BELT	Ø49	GLASGOW	Ø91	POLSON
ØØ8	BIG-SANDY	Ø5Ø	GLENDIVE	Ø92	POPLAR
ØØ9	BIG-TIMBER	Ø51	GRASSRANGE	Ø93	RED-LODGE
Ø1Ø	BILLINGS	Ø52	GREAT-FALLS	Ø94	REXFORD
Ø11	BOULDER	Ø53	HAMILTON	Ø95	RICHEY
Ø12	BOZEMAN	Ø54	HARDIN	Ø96	RONAN
Ø13	BRIDGER	Ø55	HARLEM	Ø97	ROUNDUP
Ø14	BROADUS	Ø56	HARLOWTON	Ø98	RYEGATE
Ø15	BROADVIEW	Ø57	HAVRE	Ø99	SACO
Ø16	BROCKTON	Ø58	HELENA	1ØØ	ST-IGNATIUS
Ø17	BROWNING	Ø59	HINGHAM	1Ø1	SCOBEY
Ø18	BUTTE	Ø6Ø	HOBSON	1Ø2	SHELBY
Ø19	CASCADE	Ø61	HOT-SPRINGS	1Ø3	SHERIDAN
Ø2Ø	CHESTER	Ø62	HYSHAM	1Ø4	SIDNEY
Ø21	CHINOOK	Ø63	ISMAY	1Ø5	STANFORD
Ø22	CHOTEAU	Ø64	JOLIET	1Ø6	STEVENSVILLE
Ø23	CIRCLE	Ø65	JORDAN	1Ø7	SUNBURST
Ø24	CLYDE-PARK	Ø66	JUDITH-GAP	1Ø8	SUPERIOR
Ø25	COLUMBIA-FALLS	Ø67	KALISPELL	1Ø9	TERRY
Ø26	COLUMBUS	Ø68	KEVIN	11Ø	THOMPSON-FALLS
Ø27	CONRAD	Ø69	LAUREL	111	THREE-FORKS
Ø28	CULBERTSON	Ø7Ø	LAVINA	112	TOWNSEND
Ø29	CUT-BANK	Ø71	LEWISTOWN	113	TROY
Ø3Ø	DARBY	Ø72	LIBBY	114	TWIN-BRIDGES
Ø31	DEER-LODGE	Ø73	LIMA	115	VALIER
Ø32	DENTON	Ø74	LIVINGSTON	116	VIRGINIA-CITY
Ø33	DILLON	Ø75	LODGE-GRASS	117	WALKERVILLE
Ø34	DODSON	Ø76	MALTA	118	WESTBY
Ø35	DRUMMOND	Ø77	MANHATTAN	119	WEST-YELLOWSTONE
Ø36	DUTTON	Ø78	MEDICINE-LAKE	12Ø	WHITEFISH
Ø37	EAST-HELENA	Ø79	MELSTONE	121	WHITEHALL
Ø38	EKALAKA	Ø8Ø	MILES-CITY	122	WH-SULPHUR-SPRINGS
Ø39	ENNIS	Ø81	MISSOULA	123	WIBAUX
Ø4Ø	EUREKA	Ø82	MOORE	124	WINIFRED
Ø41	FAIRFIELD	Ø83	NASHUA	125	WINNETT
Ø42	FAIRVIEW	Ø84	NEIHART	126	WOLF-POINT

TABLE 2-2

NAMES CODED IN COUNTY PARAMETER

County Number	Name as Coded	County Number	Name as Coded	County Number	Name as Coded
Ø1	SILVER-BOW	2Ø	VALLEY	38	GLACIER
Ø2	CASCADE	21	TOOLE	39	FALLON
Ø3	YELLOWSTONE	22	BIG-HORN	4Ø	SWEET-GRASS
Ø4	MISSOULA	23	MUSSELSHELL	41	MCCONE
Ø5	LEWIS-AND-CLARK	24	BLAINE	42	CARTER
Ø6	GALLATIN	25	MADISON	43	BROADWATER
Ø7	FLATHEAD	26	PONDERA	44	WHEATLAND
Ø8	FERGUS	27	RICHLAND	45	PRAIRIE
Ø9	POWDER-RIVER	28	POWELL	46	GRANITE
1Ø	CARBON	29	ROSEBUD	47	MEAGHER
11	PHILLIPS	3Ø	DEER-LODGE	48	LIBERTY
12	HILL	31	TETON	49	PARK
13	RAVALLI	32	STILLWATER	5Ø	GARFIELD
14	CUSTER	33	TREASURE	51	JEFFERSON
15	LAKE	34	SHERIDAN	52	WIBAUX
16	DAWSON	35	SANDERS	53	GOLDEN-VALLEY
17	ROOSEVELT	36	JUDITH-BASIN	54	MINERAL
18	BEAVERHEAD	37	DANIELS	55	PETROLEUM
19	CHOUTEAU			56	LINCOLN

Note: The county numbers listed in this table are based on the vehicle registration numbering system. In most other HIS subsystems, the alphabetical numbering system is used.

The MAX-#-ENTRIES Parameter - This parameter is implemented primarily as protection to the user. It can save the cost of a very expensive run when the user's input is in error or when the user's estimate of how many accidents meet certain criteria is in error. It places a limit on the number of accidents that are accepted as input (only accidents that meet all of the coded selection criteria are counted). For example, if the following command is submitted:

```
:COUNT-ACCIDENTS,START-DATE=Ø1/Ø1/76,END-DATE=Ø1/31/76,
:  MAX-#-ENTRIES=2ØØ
```

only the first 2ØØ accidents for the month of January, 1976 are included in the summary. Any number from 1 to 99,999 can be specified (do not code the comma).

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters provide a flexible means of selecting accidents by route system, route number, and milepoint. They are described in detail in chapter 1 of this manual.

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting accidents. Whenever possible, use the other parameters described in this section to select accidents - they provide the cheapest selection methods. The SELECT-DD parameter provides selection ability based on most of the data elements in the accident, roadlog, and traffic files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on the use of SELECT-DD and SELECT-SIZE is included in chapter 12. When using the select subsystem in conjunction with accident commands, the following should be kept in mind:

1. If your command does not contain a DATA parameter, you can select only on the accident detail and vehicle files. If you select on the vehicle file, your job cost can be expected to be high.

2. If your command contains a DATA parameter, you can select from the accident detail, vehicle, and directory files and on the roadlog and traffic files. Selection on any files other than the directory file can raise the job cost substantially.

The LIST Program

The LIST program prints a listing of accidents selected by means of command parameters. The listing can be in either of two formats:

1. Abbreviated format - Prints one line per accident showing the accident number, date occurred, time occurred, city number, county number, and location.
2. Complete format - Prints all data items except sensitive information such as driver name (this format requires about one page per accident printed).

With either format, a small summary showing the number of accidents, number of injury accidents, number of fatality accidents, number of injuries, and number of fatalities is printed following the listing.

To use LIST, submit a command that specifies the program LIST and that includes the parameter FILE=ACCIDENT. Use the following parameter to select one of the two list formats:

$$\text{LIST=} \left\{ \begin{array}{l} \text{FORMATTED} \\ \underline{\text{NOT-FORMATTED}} \end{array} \right\}$$

LIST=NOT-FORMATTED is the default if the LIST parameter is omitted, and requests the abbreviated listing. LIST=FORMATTED requests the complete listing.

Any of the following parameters, which are described earlier in this chapter in the section "Commands - Accident Subsystem," can be coded on LIST commands:

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION, CITY, and COUNTY
MAX-#-ENTRIES
DATA, START-MILEPOINT, and END-MILEPOINT
SELECT-DD and SELECT-SIZE

Any of the print parameters described in chapter 1 can be coded on LIST commands.

Figure 2-1 shows a sample run of LIST using the abbreviated format. The job setup used to obtain this output is:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:LIST,FILE=ACCIDENT,LIST=NOT-FORMATTED,MAX-#-ENTRIES=2Ø,
:           START-ACCIDENT=74Ø52ØØØØØØØ
/*
```

Figure 2-2 shows a sample run of LIST using the complete format. The job setup used to obtain this output is:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:LIST,FILE=ACCIDENT,LIST=FORMATTED,MAX-#-ENTRIES=1,
:           START-ACCIDENT=74Ø52ØØØØØØØ
/*
```

The following files are accessed by the LIST program:

DD Statement	File
ACIDENT	Accident detail file - Always accessed
ACCVEH	Accident vehicle file - Accessed if LIST=FORMATTED or if a select statement refers to the vehicle file.
ACCDIRI	Accident directory file - Accessed if DATA is coded.
ROADLOG	Roadlog file - Accessed if a select statement refers to the roadlog file.
TRAFFIC	Traffic file - Accessed if a select statement refers to the traffic file.
TABLES	Tables - Accessed if a select statement is included.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

```
HIS ACCIDENT FILE LISTING

740520020317  02/05/74  10:35  052  02  M05214030634
740520020392  02/20/74  12:57  052  02  P01113100814
740520020457  03/07/74  07:54  052  02  M05208430353
740520020500  03/15/74  10:48  052  02  M05211460644
740520021391  08/19/74  18:02  052  02  P01112000812
740520022124  12/22/74  13:10  052  02  P01111470795
740520040051  01/06/74  03:14  052  02  M05211180221
740520040072  01/07/74  23:21  052  02  M05208740252
740520040259  02/01/74  00:05  052  02  P05608550604
740520040288  02/02/74  15:36  052  02  P05611200610
740520040303  02/04/74  20:43  052  02  M05210650693
740520040318  02/05/74  12:55  052  02  P01115210817
740520040351  02/09/74  20:15  052  02  M05210040654
740520040352  02/09/74  21:25  052  02  M05211100693
740520040361  02/12/74  12:18  052  02  M05212520862
740520040363  02/13/74  07:50  052  02  M05214870748
740520040389  02/19/74  10:00  052  02  U20710380660
740520040437  03/03/74  10:55  052  02  P05608550605
740520040444  03/05/74  09:59  052  02  P04009220440
740520040482  03/13/74  06:50  052  02  P01113150815

***** ACCRD-100 -- MAX-#-ENTRIES COMPLETED *****


                               ----- ACCIDENT STATISTICS -----

                                     NUMBER OF ACCIDENTS          20
                              NUMBER OF FATALITY ACCIDENTS         0
                                NUMBER OF INJURY ACCIDENTS         3
                                     NUMBER OF FATALITIES          0
                                       NUMBER OF INJURIES          5
```

Figure 2-1. Accident LIST Program - LIST=NOT-FORMATTED.

```
HIS ACCIDENT FILE LISTING

740520020317  02/05/74  10:35  052  02  M05214030634

    *****  ACCIDENT DETAILS  *****

        FIRST HARMFUL EVENT........  4        DAY OF WEEK.........   TUESDAY        LEGALLY REPORTABLE........  NO
        FIRST OBJECT HIT........... 24        NUMBER OF VEHICLES.........  2        INVESTIGATED.............. YES
        INJURY SEVERITY............  0        NUMBER OF PEDESTRIANS......  0        ENGINEERING STUDY.........  NO
        DAMAGE SEVERITY............  3        NUMBER OF FATALITIES.......  0        OTHER DAMAGE -- TYPE.......  0
        CLASS OF TRAFFICWAY........  8        NUMBER OF INJURIES.........  0        OTHER DAMAGE -- SEVERITY...  0
        ROADWAY-RELATED LOCATION...  1        ANALYSIS -- FIELD 1........  0        OTHER DAMAGE -- OWNER......  0
        JUNCTION-RELATED LOCATION..  3        ANALYSIS -- FIELD 2........  0        DATE NOTIFIED........  0/00/00
        WEATHER CONDITION..........  3        COLLISION TYPE.............  6        TIME NOTIFIED........     0:00
        ROAD CONDITION.............  3        TRAFFIC CONTROLS...........  0        DATE ARRIVED.........  0/00/00
        LIGHT CONDITION............  1        POSTED SPEED LIMIT......... 15        TIME ARRIVED.........     0:00

    ***** VEHICLE NUMBER 01 *****

        STATE OF DRIVER LICENSE.... MT        CHARGE CODE............ 000000        RE-EXAMINATION RECOMMENDED..........  NO
        VISION OBSTRUCTIONS......... 0        INTENT.....................  1        INVOLVED IN INTERSTATE TRAFFIC......  NO
        PHYSICAL DEFECTS............ 0        BODY STYLE.................  1        VEHICLE YEAR........................  66
        POSSIBLE VIOLATION.......... 0        TRAILER STYLE..............  0        VEHICLE LICENSE......... 000002-6105MT73
        ROAD DEFECTS................ 0                                              DAMAGE SEVERITY.......................  3
        MECHANICAL DEFECTS.......... 0                                              DAMAGE LEVEL............      UNDER $250

                          ALCOHOL   AGE   SEX   INJURY

             DRIVER          0       73    M      0
             FRONT CENTER    0       00    0      0
             FRONT RIGHT     0       00    0      0
             REAR LEFT       0       00    0      0
             REAR CENTER     0       00    0      0
             REAR RIGHT      0       00    0      0

    ***** VEHICLE NUMBER 02 *****

        STATE OF DRIVER LICENSE.... MT        CHARGE CODE............ 05690H        RE-EXAMINATION RECOMMENDED..........  NO
        VISION OBSTRUCTIONS......... 0        INTENT.....................  9        INVOLVED IN INTERSTATE TRAFFIC......  NO
        PHYSICAL DEFECTS............ 0        BODY STYLE.................  5        VEHICLE YEAR........................  72
        POSSIBLE VIOLATION.......... 6        TRAILER STYLE..............  0        VEHICLE LICENSE......... 00002T10175MT73
        ROAD DEFECTS................ 0                                              DAMAGE SEVERITY.......................  0
        MECHANICAL DEFECTS.......... 0                                              DAMAGE LEVEL............      UNDER $250

                          ALCOHOL   AGE   SEX   INJURY

             DRIVER          0       62    M      0
             FRONT CENTER    0       00    0      0
             FRONT RIGHT     0       00    0      0
             REAR LEFT       0       00    0      0
             REAR CENTER     0       00    0      0
             REAR RIGHT      0       00    0      0
-----------------------------------------------------------------------------------------------------------------
```

Figure 2-2. Accident LIST Program - LIST=FORMATTED.

The LIST-FA-ACC-DIREC Program

The LIST-FA-ACC-DIREC (List Federal Aid Accident Directory) prints accidents that are stored in the accident directory file. The listing can be in either of two formats:

1. Abbreviated format - Prints data from directory file only (one line per accident).
2. Full format - Prints the data included in the appreviated format plus certain items from the accident detail file (one line per accident). This format costs substantially more money on long runs.

The following data items are printed in the abbreviated format:

Route system, route number, and milepoint
Accident number
Date occurred
Number of fatalities
Number of injuries
First harmful event
Road condition
Collision type

The following additional items are printed in the full format:

County number
Time occurred
Number of vehicles
Number of pedestrians
Junction-related location
Roadway-related location
Weather condition
Light condition

If LOCATION=STATEWIDE, COUNTY, or a select statement that refers to the detail file is specified, the full format is always printed. Note that only rural on-system accidents can be printed by LIST-FA-ACC-DIREC because these are the only accidents that have entries in the accident directory file.

To use LIST-FA-ACC-DIREC, prepare a command that specifies the LIST-FA-ACC-DIREC program. A DATA parameter must be included on the command. Any of the following selection parameters, which are described earlier in this chapter in the section "Commands - Accident Subsystem," can be coded on the command:

START-MILEPOINT and END-MILEPOINT
START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION or COUNTY
SELECT-DD and SELECT-SIZE
MAX-#-ENTRIES

Use the following parameter to select one of the two output formats:

$$\text{LIST=}\left\{\begin{array}{l}\text{FULL}\\ \underline{\text{ABBREVIATED}}\end{array}\right\}$$

LIST=ABBREVIATED is assumed by default if the LIST parameter is omitted from the command.

LIST-FA-ACC-DIREC normally prints a header page that describes the codes that appear in the listing. This header page can be a nuisance, especially when the listing is being obtained from a CRJE terminal. The header page can be controlled with the following parameter:

$$\text{CODE-LIST=}\left\{\begin{array}{l}\text{NO}\\ \underline{\text{YES}}\end{array}\right\}$$

CODE-LIST=YES is the default and requests that the header page be printed.

Any of the print parameters described in chapter 1 can be coded on LIST-FA-ACC-DIREC commands.

Figure 2-3 shows the header page that is printed unless CODE-LIST=NO is specified. Figure 2-4 shows a sample abbreviated format. These two pages were obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:LIST-FA-ACC-DIREC,LIST=ABBREVIATED,CODE-LIST=YES,
:    DATA=PRIMARY=8,START-DATE=Ø1/Ø1/74,END-DATE=12/31/75,
:    MAX-#-ENTRIES=25
/*
```

Figure 2-5 shows a sample full format. This listing was obtained with the following job setup:

LIST-FA-ACC-DIREC

```
FIELDS LISTED ARE:

   ROUTE NUMBER           NUMBER OF VEHICLES       JUNCTION-RELATED LOCATION
   MILEPOINT              NUMBER OF PEDESTRIANS    ON/OFF ROADWAY
   COUNTY                 NUMBER OF FATALITIES     WEATHER CONDITION
   ACCIDENT NUMBER        NUMBER OF INJURIES       ROAD CONDITION
   DATE OF OCCURRENCE     FIRST HARMFUL EVENT      LIGHT CONDITION
   TIME OF OCCURRENCE                              COLLISION TYPE


*-------- ACCIDENT NUMBER --------*

   *--- INVESTIGATED ---*                  *--- UNINVESTIGATED ---*
    1-2   YEAR                              1-2   YEAR
    3-5   AGENCY                              3   R=RURAL, M=MUNICIPAL
    6-8   BADGE NUMBER                      4-6   COUNTY (RURAL) OR CITY (MUNICIPAL)
    9-10  MONTH                             7-8   MONTH
   11-12  SEQUENCE NUMBER                   9-12  SEQUENCE NUMBER


*-------- FIRST HARMFUL EVENT --------*

  00  NOT STATED                              06  COLLISION WITH PARKED MV
  01  OVERTURNED                              07  COLLISION WITH RAILWAY TRAIN
  02  OTHER NON-COLLISION                     08  COLLISION WITH PEDALCYCLE
  03  COLLISION WITH PEDESTRIAN               09  COLLISION WITH ANIMAL
  04  COLLISION WITH MV IN TRANSPORT          10  COLLISION WITH FIXED OBJECT
  05  COLLISION WITH MV IN OTHER ROADWAY      11  COLLISION WITH OTHER OBJECT


*-------- JUNCTION-RELATED LOCATION AND ON/OFF ROADWAY --------*

         ----- JUNCTION -----                        ----- ON/OFF -----
  0  NON-JUNCTION    2  INTERSECTION-RELATED          1  ON ROADWAY
  1  INTERSECTION    3  DRIVEWAY ACCESS               2  OFF ROADWAY


*-------- WEATHER CONDITION AND ROAD CONDITION --------*

        ----- WEATHER -----                ----- ROAD -----
  0  NOT STATED    3  SNOWING          0  NOT STATED    3  SNOWY
  1  CLEAR         4  FOG              1  DRY           4  ICY
  2  RAINING       5  OTHER            2  WET           5  OTHER


*-------- LIGHT CONDITION AND COLLISION TYPE --------*

        ----- LIGHT -----                        ----- COLLISION -----
  0  NOT STATED    3  DARKNESS,LIT          1  HEAD ON    4  SIDESWIPE MEETING
  1  DAYLIGHT      4  DARKNESS,UNLIT        2  REAR END   5  SIDESWIPE PASSING
  2  DAWN/DUSK     5  OTHER                 3  ANGLE      6  BACKED INTO
                                                          7  OTHER
```

Figure 2-3. LIST-FA-ACC-DIREC Program - Code List.

ROUTE NUMBER	MILEPOINT	CNTY	ACCIDENT NUMBER	DATE	TIME	# VEH	# PED	# FAT	# INJ	EVNT	JUNCT	ONOFF	WEATH	ROAD	LIGHT	TYPE
P00008	001+0.100		750002190204	2/22/75						1				4		7
P00008	001+0.500		74R02812A001	12/01/74						4				1		4
P00008	002+0.000		740001460901	9/07/74						9				1		7
P00008	002+0.300		740003281101	11/01/74						1				2		7
P00008	002+0.300		740003281203	12/06/74						1				4		7
P00008	002+0.400		750003281102	11/17/75						1				4		7
P00008	003+0.100		750002190102	1/08/75					1	4				3		5
P00008	004+0.200		740002761102	11/05/74					1	1				1		7
P00008	004+0.500		750001600502	5/07/75						4				1		5
P00008	005+0.200		750002191107	11/30/75						10				4		7
P00008	005+0.300		750001601002	10/07/75					1	1				2		7
P00008	005+0.400		740001571003	10/22/74					2	10				1		7
P00008	006+0.700		740002190701	7/04/74						10				1		7
P00008	007+0.200		740001600102	1/16/74						1				1		7
P00008	007+0.400		740002191103	11/28/74						4				4		4
P00008	007+0.400		750003281002	10/07/75					1	1				2		7
P00008	007+0.600		750002191007	10/26/75						1				4		7
P00008	008+0.500		740003070603	6/10/74				1	1	1				1		7
P00008	008+0.600		750002191105	11/21/75						10				1		7
P00008	008+0.700		750002190801	8/01/75				1	1	1				1		4
P00008	008+0.800		750002191003	10/11/75					1	10				2		7
P00008	011+0.900		740002190808	8/22/74					2	10				1		7
P00008	012+0.900		750002191102	11/07/75						1				4		7
P00008	013+0.000		74R02812A002	12/06/74						4				3		2
P00008	014+0.070		75R02809A002	9/13/75						10				1		7

***** ACCRD-100 -- MAX-#-ENTRIES COMPLETED *****

---- NUMBER OF RECORDS PRINTED: 25

Figure 2-4. LIST-FA-ACC-DIREC Program - LIST=ABBREVIATED.

ROUTE NUMBER	MILEPOINT	CNTY	ACCIDENT NUMBER	DATE	TIME	# VEH	# PFD	# FAT	# INJ	EVNT	JUNCT	ONOFF	WEATH	ROAD	LIGHT	TYPE
P00008	001+0.100	28	750002190204	2/22/75	1900	1				1		2	1	4	4	7
P00008	001+0.500	28	74R02812A001	12/01/74	0000	2				4		1	1	1	4	4
P00008	002+0.000	28	740001460901	9/07/74	2250	1				9		1	1	1	4	7
P00008	002+0.300	28	740003281101	11/01/74	2030	1				1		2	2	2	3	7
P00008	002+0.300	28	740003281203	12/06/74	2150	1				1		2	3	4	4	7
P00008	002+0.400	28	750003281102	11/17/75	1050	1				1		2	1	4	1	7
P00008	003+0.100	25	750002190102	1/08/75	1630	2			1	4		1	1	3	1	5
P00008	004+0.200	28	740002761102	11/05/74	0230	1			1	1		2	1	1	4	7
P00008	004+0.500	28	750001600502	5/07/75	1320	2				4		1	1	1	1	5
P00008	005+0.200	28	750002191107	11/30/75	1930	1				10		2	3	4	4	7
P00008	005+0.300	28	750001601002	10/07/75	1530	1			1	1		2	2	2	1	7
P00008	005+0.400	28	740001571003	10/22/74	1815	1			2	10		2	1	1	1	7
P00008	006+0.700	28	740002190701	7/04/74	1145	1				10		2	1	1	1	7
P00008	007+0.200	28	740001600102	1/16/74	1930	1				1		2	1	1	4	7
P00008	007+0.400	28	740002191103	11/28/74	0345	2				4		1	1	4	4	4
P00008	007+0.400	28	750003281002	10/07/75	1300	1			1	1		2	2	2	1	7
P00008	007+0.600	28	750002191007	10/26/75	1930	1				1		2	3	4	4	7
P00008	008+0.500	28	740003070603	6/10/74	1835	1		1	1	1		2	1	1	1	7
P00008	008+0.600	28	750002191105	11/21/75	1345	1				10		2	1	1	1	7
P00008	008+0.700	28	750002190801	8/01/75	0920	2		1	1	1		1	1	1	1	4
P00008	008+0.800	28	750002191003	10/11/75	1600	1			1	10		2	2	2	1	7
P00008	011+0.900	28	740002190803	8/22/74	0220	1			2	10		2	1	1	4	7
P00008	012+0.900	28	750002191102	11/07/75	0700	1				1		2	2	4	2	7
P00008	013+0.000	28	74R02812A002	12/06/74	1820	2				4		1	3	3	4	2
P00008	014+0.070	28	75R02809A002	9/13/75	0100	1				10		2	1	1	3	7

***** ACCRD-100 -- MAX-#-ENTRIES COMPLETED *****

---- NUMBER OF RECORDS PRINTED: 25

Figure 2-5. LIST-FA-ACC-DIREC Program - LIST=FULL.

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:LIST-FA-ACC-DIREC,LIST=FULL,CODE-LIST=NO,
:    DATA=PRIMARY=8,START-DATE=Ø1/Ø1/74,END-DATE=12/31/75,
:    MAX-#-ENTRIES=25
/*
```

The following files are accessed by the LIST-FA-ACC-DIREC program:

DD Statement	File
ACCDIRI	Accident directory file - always accessed
ACIDENT	Accident detail file - Accessed for LIST=FULL, COUNTY=county, LOCATION=STATEWIDE, or a select statement that refers to the detail file.
ACCVEH	Accident vehicle file - Accessed if a select statement refers to the vehicle file.
ROADLOG	Roadlog file - Accessed if a select statement refers to the roadlog file.
TRAFFIC	Traffic file - Accessed if a select statement refers to the traffic file.
TABLES	Tables - Accessed if (1) a COUNTY parameter is coded, (2) a select statement is included, or (3) CODE-LIST=YES is specified or assumed by default.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

The COUNT-ACCIDENTS Program

COUNT-ACCIDENTS prints a small summary of accident data. The summary shows the number of property damage, fatality, and injury accidents and the number of injuries and fatalities. To use the program, prepare a command that specifies the COUNT-ACCIDENTS program. Any of the following selection parameters can be included on the command (these parameters are described earlier in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION, CITY, or COUNTY
DATA, START-MILEPOINT, AND END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and SELECT-SIZE

```
HIS ACCIDENT SUMMARY TOTALS

 TOTAL NUMBER OF ACCIDENTS                     17,783

 NUMBER OF FATAL ACCIDENTS                        247          1.39 PERCENT OF TOTAL
 NUMBER OF INJURY ACCIDENTS                     5,545         31.18 PERCENT OF TOTAL
    SUB-TOTAL: FATAL + INJURY                   5,792             ( 32.57  PER CENT)

 NUMBER OF PROPERTY DAMAGE ACCIDENTS           11,991         67.43 PERCENT OF TOTAL

 NUMBER OF FATALITIES                             299
 NUMBER OF INJURIES                             8,466
```

Figure 2-6. COUNT-ACCIDENTS Program.

In addition, any of the print parameters described in chapter 1 can be included on the command.

Figure 2-6 shows a sample COUNT-ACCIDENTS output. This run was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:COUNT-ACCIDENTS,START-DATE=Ø1/Ø1/74,END-DATE=12/31/74
/*
```

The following files are accessed by the COUNT-ACCIDENTS program:

DD Statement	File
ACIDENT	Accident detail file - always accessed.
ACCVEH	Accident vehicle file - Accessed if a select statement refers to the vehicle file.
ACCDIRI	Accident directory file - Accessed if a DATA parameter is coded.
ROADLOG	Roadlog file - Accessed if a select statement refers to the roadlog file.
TRAFFIC	Traffic file - Accessed if a select statement refers to the traffic file.
TABLES	Tables - Accessed if CITY, COUNTY, or SELECT-DD is coded.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

The SUM-BY-DAY-&-TIME Program

SUM-BY-DAY-&-TIME prints one or more summaries broken down by time of day and by day of week. When a CITY parameter is coded, only one summary is printed. Otherwise, three summaries are printed: one of rural accidents, one of municipal accidents, and one of the combined totals.

To use SUM-BY-DAY-&-TIME, prepare a command that specifies the SUM-BY-DAY-&-TIME program. Any of the following selection parameters can be included on the command (these parameters are described earlier in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION, CITY, or COUNTY
DATA, START-MILEPOINT, and END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and SELECT-SIZE

Any of the print parameters described in chapter 1 can be included on the command.

Figure 2-7 shows the last of three pages printed by the SUM-BY-DAY-&-TIME program with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:SUM-BY-DAY-&-TIME,START-DATE=Ø1/Ø1/74,END-DATE=12/31/74
/*
```

SUM-BY-DAY-&-TIME accesses the same files under the same conditions as shown above for COUNT-ACCIDENTS.

The SUM-BY-CONTR-CIRC Program

SUM-BY-CONTR-CIRC prints a set of summaries broken down by contributing circumstances. The breakdowns are based on the analysis fields stored in the accident detail file. Up to six summaries are printed showing the following types of location:

1. Federal aid interstate system.
2. Federal aid primary system.
3. Federal aid secondary system.
4. Federal aid urban system.
5. Not on federal aid system.
6. Combined total of all systems.

Each of the six summaries requires one printed page and each is printed in the same format. Within each summary, the accidents are broken down as rural (outside the city limits of incorporated cities) or urban (inside the city limits of incorporated cities). A total line provides a count of the number of accidents selected for the summary, but this total line is not necessarily the total of the accidents shown in the summary (some accidents do not appear at all in the summary because they do not have an analysis field coded, and some appear twice because two analysis fields can be stored for one accident).

ALL STATEWIDE ACCIDENTS FROM 01/01/74 TO 12/31/74

TOTAL
*****

HOUR	TOTAL ALL	TOTAL FATAL	MONDAY ALL	MONDAY FATAL	TUESDAY ALL	TUESDAY FATAL	WEDNESDAY ALL	WEDNESDAY FATAL	THURSDAY ALL	THURSDAY FATAL	FRIDAY ALL	FRIDAY FATAL	SATURDAY ALL	SATURDAY FATAL	SUNDAY ALL	SUNDAY FATAL
MIDNIGHT	595	22	54	1	50	2	56	2	56	1	88	3	169	10	122	3
1:00	710	15	60	1	60	1	61		76	5	70	1	206	4	177	3
2:00	815	13	71	1	68	3	75	1	66	2	81	1	219	6	235	1
3:00	395	11	27	1	40	1	26		43	1	41	1	109	3	109	4
4:00	206	5	22	1	20		15		23		29	1	49	2	48	1
5:00	155	4	24	1	17		12		17	1	16	1	38	1	31	
6:00	226	1	38		33		37	1	25		29		29		35	
7:00	491	11	80	2	89	1	83	2	88	1	72	2	45	2	34	1
8:00	580	3	90		90	1	99		102	1	98		69	1	32	
9:00	527	3	97		80	2	85		61		69	1	93		42	
10:00	650	7	90	1	90	2	87		104	1	103		93	1	83	2
11:00	801	8	120	3	115	1	112	3	114		130	1	135		75	
NOON	909	7	123	2	129	1	140		119	1	142	1	153	1	103	1
1:00	840	6	123	1	119		128	1	124	2	128	1	124	1	94	
2:00	925	6	130	1	140	1	106		121	1	167	1	136	1	125	1
3:00	1159	19	172	1	169	1	158	5	149	2	212	4	177	5	122	1
4:00	1242	11	177	1	180	2	181		156	3	209	2	194	2	145	1
5:00	1252	14	156	3	191	1	207	2	176	2	202	3	180	3	138	
6:00	923	18	94	6	101		115	2	121	1	171	3	188	4	136	2
7:00	942	11	126	2	98	2	126	1	128		173	1	156	3	135	2
8:00	900	10	98		93	2	112	2	117	1	136	2	129	2	115	1
9:00	810	10	94	2	103	3	109	2	102		172	1	133	1	97	1
10:00	837	19	78		89	3	116	2	110	4	181	2	165	6	98	2
11:00	827	10	72	1	99	1	90		98		184	2	198	5	86	1
NOT STATED	161	1	24		16		14		16		31		39	1	21	
TOTALS	17783	247	2240	32	2279	31	2350	26	2314	30	2934	35	3226	65	2440	28

Figure 2-7. SUM-BY-DAY-&-TIME Program.

To use SUM-BY-CONTR-CIRC, prepare a command that specifies the SUM-BY-CONTR-CIRC program. Any of the following selection parameters can be included on the command (these parameters are described earlier in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION, CITY, or COUNTY
DATA, START-MILEPOINT, and END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and SELECT-SIZE

Any of the print parameters described in chapter 1 can be included on the command.

Figure 2-8 shows the last page printed by a SUM-BY-CONTR-CIRC command. The run was produced with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:SUM-BY-CONTR-CIRC,START-DATE=Ø1/Ø1/74,END-DATE=12/31/74
/*
```

SUM-BY-CONTR-CIRC accesses the same files under the same circumstances as shown above for COUNT-ACCIDENTS.

The FORM-16 Program

FORM-16 produces 19 of the 21 summaries that comprise the National Safety Council's Form 16 report. Each FORM-16 command prints three sets of the 19 summaries. If a CITY parameter is coded on the command the following three sets of summaries are printed:

1. Legally reportable accidents.
2. Non-reportable accidents.
3. Combined totals of reportable and non-reportable accidents.

Otherwise the following three sets of summaries are printed:

1. Rural accidents.
2. Municipal accidents.
3. Combined totals of rural and municipal accidents.

SUMMARY BY CONTRIBUTING CIRCUMSTANCES

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 01/01/74 TO 12/31/74

ALL SYSTEMS	RURAL NUMBER OF ACCIDENTS	RURAL PERCENTAGE OF ACCIDENTS	URBAN NUMBER OF ACCIDENTS	URBAN PERCENTAGE OF ACCIDENTS	TOTAL NUMBER OF ACCIDENTS	TOTAL PERCENTAGE OF ACCIDENTS
FAILED TO HAVE VEHICLE UNDER CONTROL	922	9.78	54	.65	976	5.49
INATTENTIVE DRIVING	1,591	16.88	116	1.39	1,707	9.60
INEXPERIENCE	324	3.44	20	.24	344	1.93
BLACKOUT, HEART, STROKE, ETC	43	.46	22	.26	65	.37
FELL ASLEEP	562	5.96	40	.48	602	3.39
SUN GLARE	55	.58	41	.49	96	.54
RAINING	44	.47	4	.05	48	.27
SNOWING	53	.56	3	.04	56	.31
WHITEOUT	16	.17			16	.09
BLOWING SNOW	80	.85	5	.06	85	.48
WHITEOUT -- MEETING OR FOLLOWING VEH	33	.35	1	.01	34	.19
DUST STORM	5	.05			5	.03
DUST CAUSED BY WIND OR VEHICLE	20	.21	1	.01	21	.12
ROAD SLIPPERY OR ICY	915	9.71	279	3.34	1,194	6.71
OTHER WEATHER CONDITIONS	38	.40	2	.02	40	.22
IMPROPER HITCH	47	.50	4	.05	51	.29
BLOW OUT -- FLAT TIRE	148	1.57	11	.13	159	.89
STONE THROWN BY VEHICLE	4	.04			4	.02
AVOIDING ANOTHER VEHICLE	362	3.84	114	1.36	476	2.68
AVOIDING PEDESTRIAN	18	.19	16	.19	34	.19
STRIKING OR AVOIDING DOMESTIC ANIMAL	527	5.59	42	.50	569	3.20
STRIKING OR AVOIDING WILD ANIMAL	480	5.09			480	2.70
STRIKING OR AVOIDING OBJECT IN ROAD	44	.47	2	.02	46	.26
DISTRACTION WITHIN VEHICLE	121	1.28	39	.47	160	.90
DISTRACTION FROM OUTSIDE VEHICLE	46	.49	16	.19	62	.35
UNWARRANTED SLOWING	17	.18			17	.10
BLINDED BY GLARING LIGHTS	4	.04			4	.02
PASSENGER FELL FROM VEHICLE	8	.08	1	.01	9	.05
OCCUPANT RELEASES VEHICLE	2	.02			2	.01
INDIAN IN VIOLATION ON RESERVATION	8	.08			8	.04
TRAF CONTROL SIGN -- MISSING, DOWN	8	.08	2	.02	10	.06
WIND BLOWING	89	.94	1	.01	90	.51
WATER ON HIGHWAY	39	.41	2	.02	41	.23
FOG	10	.11			10	.06
LOAD SHIFTED	71	.75	1	.01	72	.40
TOTAL NUMBER OF ACCIDENTS	9,428		8,355		17,783	

Figure 2-8. SUM-BY-CONTR-CIRC Program.

The 19 summaries printed in each set are:

1. Table 1-a - Number of accidents by first harmful event, accident severity, and on/off roadway.
2. Table 1-b - Number of persons by first harmful event and injury severity.
3. Table 2-a - Number of accidents, persons killed, and persons injured this year to date and for the same period in the previous year, broken down by first harmful event and on/off roadway.
4. Table 3-a - Number of accidents occurring in cities of population 2,5ØØ or greater broken down by population, accident severity, and on/off roadway. Also, number of persons killed or injured in cities of population 2,5ØØ or greater broken down by population.
5. Table 3-b - Number of accidents occurring within incorporated cities broken down by class of trafficway, accident severity, and on/off roadway. Also, number of persons killed or injured within incorporated cities broken down by class of trafficway.
6. Table 3-c - Number of accidents occurring outside incorporated cities (rural accidents) broken down by class of trafficway, accident severity, and on/off roadway. Also, number of persons killed or injured outside incorporated cities broken down by class of trafficway.
7. Table 4 - Number of persons, number of pedestrians, and number of pedalcyclists broken down by age and sex.
8. Table 5-a - Number of intersection accidents involving two motor vehicles broken down by accident type and severity.
9. Table 5-b - Number of non-intersection accidents involving two motor vehicles broken down by accident type and severity.
1Ø. Table 5-c - Number of pedestrian accidents broken down by accident type, accident severity, and intersection location.
11. Table 5-d - Number of accidents not appearing in 5-a through 5-c broken down by accident type, accident severity, and intersection location.
12. Table 6 - Number of pedestrians killed broken down by pedestrian intent. Also, number of pedestrians killed or injured broken down by pedestrian intent and by age.
13. Table 7 - Number of drivers broken down by age and accident severity.
14. Table 8 - Number of drivers broken down by sex and by accident severity.
15. Table 1Ø - Number of accidents involving contributing circumstances broken down by contributing circumstances and accident severity (the analysis fields of the detail file, the contributing circumstances fields of the vehicle file, and actual charge codes are used in computing this table).

16. Table 11 - Number of vehicles by body style and accident severity.

17. Table 12 - Number of accidents by road condition and accident severity.

18. Table 13 - Number of accidents by light condition and accident severity.

19. Table 14 - Number of accidents involving two motor vehicles by collision type and accident severity.

To use the program, prepare a command that specifies the FORM-16 program. Any of the following selection parameters can be included on the command (these parameters are described earlier in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION, CITY, or COUNTY
DATA, START-MILEPOINT, and END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and SELECT-SIZE

Figure 2-9 on the following pages shows one of the three sets of summaries produced by a FORM-16 command. The run was produced with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:FORM-16,START-DATE=Ø7/Ø1/74,END-DATE=Ø7/31/74
/*
```

The following files are accessed by the FORM-16 program:

DD Statement	File
ACIDENT	Accident detail file - always accessed.
ACCVEH	Accident vehicle file - always accessed.
ACCDIRI	Accident directory file - Accessed if a DATA parameter is coded.
ROADLOG	Roadlog file - Accessed if a select statement refers to the roadlog file.
TRAFFIC	Traffic file - Accessed if a select statement refers to the traffic file.
TABLES	Tables - Accessed if CITY, COUNTY, or SELECT-DD is coded.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

LEGALLY REPORTABLE ACCIDENTS ARE THOSE INVOLVING DEATH, BODILY INJURY, OR PROPERTY DAMAGE OF $250 OR MORE TO THE PROPERTY OF ONE PERSON.

1A. TYPE OF MOTOR-VEHICLE ACCIDENT.	NUMBER OF ACCIDENTS: TOTAL				ON ROADWAY				OFF ROADWAY			
	TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE
NONCOLLISION												
1. OVERTURNING	258	8	152	98	50	1	32	17	208	7	120	81
2. OTHER NONCOLLISION	21	1	8	12	12		6	6	9	1	2	6
COLLISION INVOLVING:												
3. PEDESTRIAN	22	3	19		22	3	19					
4. MV IN TRANSPORT	768	10	232	526	759	10	231	518	9		1	8
5. MV ON OTHER ROADWAY	1		1		1		1					
6. PARKED MV	131		26	105	111		16	95	20		10	10
7. RAILWAY TRAIN	1			1	1			1				
8. PEDALCYCLIST	21	1	20		20	1	19		1		1	
9. ANIMAL	66		8	58	66		8	58				
10. FIXED OBJECT	268	6	113	149	8		2	6	260	6	111	143
11. OTHER OBJECT	8	1	4	3	6	1	2	3	2		2	
12. UNKNOWN												
TOTALS	1565	30	584	952	1056	16	335	704	509	14	247	248

1B. TYPE OF MOTOR-VEHICLE ACCIDENT.	NUMBER OF PERSONS: TOTAL KILLED	TOTAL INJURED	INCAPACITATING INJURY	NON-INCAPAC. EVIDENT INJURY	POSSIBLE INJURY	NO INJURY
NONCOLLISION						
1. OVERTURNING	8	263	81	102	80	113
2. OTHER NONCOLLISION	1	8	4	3	1	17
COLLISION INVOLVING:						
3. PEDESTRIAN	3	19	7	8	4	21
4. MV IN TRANSPORT	13	395	91	174	130	1320
5. MV ON OTHER ROADWAY		2	2			
6. PARKED MV		35	6	22	7	89
7. RAILWAY TRAIN						1
8. PEDALCYCLIST	1	20	7	9	4	21
9. ANIMAL		10	4	6		62
10. FIXED OBJECT	7	158	56	66	36	162
11. OTHER OBJECT	1	9	1	6	2	5
12. UNKNOWN						
TOTALS	34	919	259	396	264	1811

Figure 2-9. FORM-16 Program. Page 1 of 7.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

2A. TYPE OF MOTOR-VEHICLE ACCIDENT.	TOTAL: THIS YEAR TO DATE ALL ACCIDENTS	PERSONS KILLED	PERSONS INJURED	TOTAL: SAME PERIOD LAST YEAR ALL ACCIDENTS	PERSONS KILLED	PERSONS INJURED	ON ROADWAY: THIS YEAR TO DATE ALL ACCIDENTS	PERSONS KILLED	PERSONS INJURED	ON ROADWAY: SAME PERIOD LAST YEAR ALL ACCIDENTS	PERSONS KILLED	PERSONS INJURED
NONCOLLISION												
1. OVERTURNING	1242	44	1046	247	13	163	233	5	204	48	2	33
2. OTHER NONCOLLISION	137	3	52	22	2	10	69	2	28	14	1	5
COLLISION INVOLVING:												
3. PEDESTRIAN	119	14	110	18		21	115	14	105	17		20
4. MV IN TRANSPORT	4801	44	2033	763	12	417	4761	44	2013	756	12	415
5. MV ON OTHER ROADWAY	1		2	7		5	1		2	7		5
6. PARKED MV	808	4	137	114	2	30	709	2	111	103		21
7. RAILWAY TRAIN	19	3	15	5		2	18	3	14	4		2
8. PEDALCYCLIST	73	4	69	24	1	25	71	4	67	24	1	23
9. ANIMAL	239	1	35	75		16	246	1	32	74		15
10. FIXED OBJECT	1785	36	866	250	7	129	70		19	30	1	13
11. OTHER OBJECT	63	1	37	14		5	50	1	25	6		1
12. UNKNOWN												
TOTALS	9337	154	4402	1539	37	846	6363	76	2020	1063	17	553

3A. MUNICIPALITIES AND INCORPORATED TOWNSHIPS	TOTAL: TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	ON ROADWAY: TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	OFF ROADWAY: TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	NUMBER OF PERSONS FAT	INJ
1. 2,500 TO 5,000	44		12	32	41		11	30	3		1	2		16
2. 5,000 TO 10,000	29	1	8	20	23		7	15	6	1	1	4	1	12
3. 10,000 TO 25,000	103		41	67	57		35	62	11		6	5		54
4. 25,000 TO 50,000	119	2	38	79	110	2	35	73	9		3	6	3	53
5. 50,000 TO 100,000	323		84	239	294		75	219	24		9	20		115
6. 100,000 TO 250,000														
7. 250,000 AND OVER														
TOTAL	623	3	183	437	565	2	163	400	58	1	20	37	4	250

3B. URBAN	TOTAL: TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	ON ROADWAY: TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	OFF ROADWAY: TOTAL	FATAL	NONFATAL INJURY	PROP DAMAGE	NUMBER OF PERSONS FAT	INJ
1. INTERSTATE SYSTEM	7		4	3	3		2	1	4		2	2		5
2. OTHER CONTROL ACCESS														
3. OTHER US ROUTE NUMBERED	233	3	73	157	215	3	68	144	18		5	13	4	98
4. OTHER STATE NUMBERED	116		36	80	100		28	72	16		8	8		47
5. OTHER MAJOR ARTERIAL														
6. COUNTY ROADS														
7. LOCAL STREETS	332	2	91	239	309	1	84	224	23	1	7	15	2	128
8. OTHER TRAFFICWAYS	9		1	8	3		1	2	6			6		1
9. NOT STATED														
TOTAL URBAN	697	5	205	487	630	4	183	443	67	1	22	44	6	279

Figure 2-9. FORM-16 Program. Page 2 of 7.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

3C. RURAL	TOTAL: TOTAL	TOTAL: FATAL	TOTAL: NONFATAL INJURY	TOTAL: PROP DAMAGE	ON ROADWAY: TOTAL	ON ROADWAY: FATAL	ON ROADWAY: NONFATAL INJURY	ON ROADWAY: PROP DAMAGE	OFF ROADWAY: TOTAL	OFF ROADWAY: FATAL	OFF ROADWAY: NONFATAL INJURY	OFF ROADWAY: PROP DAMAGE	NUMBER OF PERSONS FAT	NUMBER OF PERSONS INJ
1. INTERSTATE SYSTEM	86	2	39	45	38	1	17	20	48	1	22	25	2	68
2. OTHER CONTROL ACCESS														
3. OTHER US ROUTE NUMBERED	326	14	126	186	180	9	53	118	146	5	73	68	16	239
4. OTHER STATE NUMBERED	203	7	94	102	93	2	33	58	110	5	61	44	8	162
5. OTHER MAJOR ARTERIAL														
6. COUNTY ROADS	235	1	110	124	103		45	58	132	1	65	66	1	157
7. LOCAL STREETS														
8. OTHER TRAFFICWAYS	18	1	9	8	12		5	7	6	1	4	1	1	14
9. NOT STATED														
TOTAL RURAL	868	25	378	465	426	12	153	261	442	13	225	204	28	640

4. AGE OF CASUALTY.	KILLED: TOTAL KILLED TOTAL	KILLED: TOTAL KILLED MALE	KILLED: TOTAL KILLED FEMALE	KILLED: PEDESTRIANS TOTAL	KILLED: PEDESTRIANS MALE	KILLED: PEDESTRIANS FEMALE	KILLED: PEDALCYCLIST TOTAL	KILLED: PEDALCYCLIST MALE	KILLED: PEDALCYCLIST FEMALE	INJURED: TOTAL INJURED TOTAL	INJURED: TOTAL INJURED MALE	INJURED: TOTAL INJURED FEMALE	INJURED: PEDESTRIANS TOTAL	INJURED: PEDESTRIANS MALE	INJURED: PEDESTRIANS FEMALE	INJURED: PEDALCYCLIST TOTAL	INJURED: PEDALCYCLIST MALE	INJURED: PEDALCYCLIST FEMALE
1. 0 TO 4	1	1		1	1					19	8	11	4	3	1			
2. 5 TO 9	1		1	1		1				38	24	14	4	3	1	8	6	2
3. 10 TO 14	2	2					1	1		57	40	17	1	1		6	5	1
4. 15 TO 19	10	6	4							302	203	99	2	1	1	8	2	6
5. 20 TO 24	5	3	2	1	1					146	104	42	3	3		1	1	
6. 25 TO 34	3	3								139	88	51						
7. 35 TO 44	2	1	1							70	42	28	2	2				
8. 45 TO 54	5	4	1							48	22	26	1		1			
9. 55 TO 64	1		1							45	24	21	1		1			
10. 65 TO 74	4	1	3							29	11	18						
11. 75 & OLDER										9	4	5						
12. NOT STATED										17	6	11						
TOTALS	34	21	13	3	2	1	1	1		919	576	343	18	13	5	23	14	9

Figure 2-9. FORM-16 Program. Page 3 of 7.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

5. DIRECTION ANALYSIS.

AN ACCIDENT CONSISTING OF A SERIES OF COLLISIONS, OVERTURNING, ETC., IS CLASSIFIED ACCORDING TO THE FIRST DAMAGE OR INJURY PRODUCING EVENT; INCLUDES ON ROADWAY AND OFF ROADWAY.

5A.	TWO MOTOR-VEHICLE ACCIDENTS. AT INTERSECTION.	TOTAL	FATAL ACCIDENTS	INJURY ACCIDENTS	PROPERTY DAMAGE ACCIDENTS
1.	ENTERING AT ANGLE	256	3	86	167
2A.	SAME DIRECTION--BOTH STRAIGHT	45		11	34
2B.	SAME--ONE TURNING, ONE STRAIGHT	58	2	15	41
2C.	SAME--ONE STOPPED	50		18	32
2D.	SAME--ALL OTHERS	7			7
3A.	OPPOSITE DIRECTION--BOTH STRAIGHT	7	1	2	4
3B.	SAME--ONE LEFT TURN, ONE STRAIGHT	26		9	17
3C.	SAME--ALL OTHERS	7		2	5
4.	NOT STATED	15		1	14
	TOTALS	471	6	144	321

5B.	TWO MOTOR-VEHICLE ACCIDENTS. NOT AT INTERSECTION.	TOTAL	FATAL ACCIDENTS	INJURY ACCIDENTS	PROPERTY DAMAGE ACCIDENTS
1.	OPPOSITE DIRECTION--BOTH MOVING	68	4	30	34
2.	SAME DIRECTION--BOTH MOVING	98		28	70
3A.	ONE CAR PARKED	106		25	81
3B.	ONE CAR STOPPED IN TRAFFIC	21		7	14
4A.	ONE CAR ENTERING PARKED POSITION				
4B.	ONE CAR LEAVING PARKED POSITION	3			3
5.	DRIVEWAY ACCESS	90	1	24	65
6.	ALL OTHERS	26	1	6	19
7.	NOT STATED				
	TOTALS	412	6	120	286

5C.	PEDESTRIAN ACCIDENTS.	ALL PEDESTRIAN ACCIDENTS	***** FATAL ACCIDENTS ***** TOTAL	INTERSECTN	DRIVEWAY, NONJCT	NON-FATAL INJURY ACCIDENTS TOTAL	INTERSECTN	DRIVEWAY, NONJCT
1.	CAR GOING STRAIGHT	14	2		2	12	1	11
2.	CAR TURNING RIGHT							
3.	CAR TURNING LEFT	2				2	2	
4.	CAR BACKING	1				1		1
5.	ALL OTHERS	3				3		3
	TOTALS	20	2		2	18	3	15

Figure 2-9. FORM-16 Program. Page 4 of 7.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

5D. ALL OTHER ACCIDENTS.

	TOTAL	FATAL ACCIDENTS	INJURY ACCIDENTS	PROPERTY DAMAGE ACCIDENTS
AT INTERSECTION				
1. OTHER ROAD VEHICLE/TRAIN	29	1	12	16
2. FIXED OBJECT	25		5	20
3. OTHER OBJECT OR ANIMAL	1		1	
NONCOLLISION				
4. OVERTURNING	15		9	6
5. OTHER NONCOLLISION	1			1
NOT AT INTERSECTION				
COLLISION WITH:				
6. OTHER ROAD VEHICLE/TRAIN	21		8	13
7. FIXED OBJECT	234	6	107	126
8. OTHER OBJECT OR ANIMAL	72	1	10	61
NONCOLLISION				
9. OVERTURNING	249	7	141	91
10. OTHER NONCOLLISION	20	1	8	11
11. NOT STATED				
TOTALS	562	16	301	345

6. PEDESTRIAN ACTIONS BY AGE.

		KILLED	TOTAL	0-4	5-9	10-14	15-19	20-24	25-44	45-64	65 & OLDER	NOT STATED
			************************ AGES OF PEDESTRIANS KILLED AND INJURED ************************									
1A.	AT INTERSECTN OR IN CROSSWALK		2							2		
1B.	NOT AT INTERSECTN OR CROSSWALK	3	13	5	4		1	2	1			
2A.	WALKING IN RDWAY--WITH TRAFFIC											
2B.	SAME--AGAINST TRAFFIC											
3.	STANDING IN ROADWAY											
4.	PUSHING/WORKING ON VEH IN RDWAY											
5.	OTHER WORKING IN ROADWAY		1					1				
6.	PLAYING IN ROADWAY		1		1							
7.	OTHER IN ROADWAY		4			1	1	1	1			
8.	NOT IN ROADWAY											
9.	NOT STATED											
	TOTALS	3	21	5	5	1	2	4	2	2		

Figure 2-9. FORM-16 Program. Page 5 of 7.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

7. AGE OF DRIVER.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1. 15 & YOUNGER	68	1	42
2. 16	122	3	54
3. 17	151	4	50
4. 18 TO 19	322	6	110
5. 20 TO 24	423	8	153
6. 25 TO 34	415	6	157
7. 35 TO 44	252	5	97
8. 45 TO 54	194	4	71
9. 55 TO 64	171	2	51
10. 65 TO 74	93	4	28
11. 75 & OLDER	35		8
12. NOT STATED	77		23
TOTALS	2332	43	850

8. SEX OF DRIVER.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1. MALE	1721	34	633
2. FEMALE	610	9	211
3. NOT STATED	1		1
TOTALS	2332	43	850

10. CONTRIBUTING CIRCUMSTANCES.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1. SPEED TOO FAST	260	8	122
2. FAILED TO YIELD RIGHT OF WAY	199	5	77
3. PASSED STOP SIGN	30		8
4. DISREGARDED TRAFFIC SIGNAL			
5. DROVE LEFT OF CENTER	1		
6. IMPROPER OVERTAKING			
7. FOLLOWED TOO CLOSELY	1		
8. MADE IMPROPER TURN	6		3
9. HAD BEEN DRINKING	135	5	71
10. OTHER IMPROPER DRIVING	405	6	140
11. MECHANICAL DEFECT	96	1	45
12. OTHER	241	8	114
TOTALS	1374	33	580

Figure 2-9. FORM-1[illegible] Program. Page 6 of 7.

STATEWIDE ACCIDENTS
REPORTING PERIOD FROM 07/01/74 TO 07/31/74

SUMMARY OF MOTOR VEHICLE ACCIDENTS

ALL ACCIDENTS

11.	TYPE OF VEHICLE.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1.	PASSENGER CAR	1683	25	510
2.	PASSENGER CAR & TRAILER	23	2	3
3.	TRUCK OR TRUCK TRACTOR	522	7	203
4.	TRUCK TRACTOR & SEMI-TRAILER	29	2	12
5.	OTHER TRUCK COMBINATION	44		8
6.	FARM TRACTOR AND/OR EQUIP	5		
7.	TAXICAB			
8.	BUS	3		2
9.	SCHOOL BUS	1		
10.	MOTORCYCLE	110	5	95
11.	MOTOR SCOOTER/MOTOR BIKE			
12.	OTHER	100	2	51
13.	NOT STATED	31		8
	TOTALS	2551	43	892
	SPECIAL VEHICLES INCLUDED ABOVE			
14.	EMERGENCY (INCL. PRIVATE)			
15.	MILITARY VEHICLES			
16.	OTHER PUBLICLY OWNED VEHS			

12.	ROAD SURFACE CONDITION.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1.	DRY	1462	27	548
2.	WET	83	3	29
3.	SNOWY OR ICY	1		
4.	OTHER	6		4
5.	NOT STATED	13		2
	TOTALS	1565	30	583

13.	LIGHT CONDITION.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1.	DAYLIGHT	1040	20	405
2.	DAWN OR DUSK	70		19
3.	DARKNESS	440	10	156
4.	NOT STATED	15		3
	TOTALS	1565	30	583

14.	MANNER OF TWO MOTOR VEHICLE COLLISION.	ALL ACCIDENTS	FATAL ACCIDENTS	INJURY ACCIDENTS
1.	HEAD ON	51	3	28
2.	REAR END	208	1	71
3.	ANGLE	298	3	97
4.	SIDESWIPE-MEETING	70	2	22
5.	SIDESWIPE-PASSING	110	2	18
6.	BACKED INTO	25		2
7.	OTHER	122	2	26
	TOTALS	884	13	264

Figure 2-9. FORM-16 Program. Page 7 of 7.

DD Statement	File
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

The SUM-BY-TRAFFICWAY Program

SUM-BY-TRAFFICWAY prints a comprehensive set of summaries broken down by class of trafficway. Each summary is broken down by the following road classes:

1. Federal aid interstate.
2. U. S. Highway.
3. State highway.
4. County road.
5. Municipal street.
6. Combined total of all systems.

Each SUM-BY-TRAFFICWAY command prints the following set of 41 summaries broken down by class of trafficway:

1. Number of accidents by county.
2. Number of accidents by first harmful event.
3. Number of accidents by severity.
4. Number of accidents by month.
5. Number of accidents by day of week.
6. Number of accidents by hour of occurrence.
7. Number of accidents by AM or PM.
8. Number of accidents by weather condition.
9. Number of accidents by road condition.
1∅. Number of accidents by light condition.
11. Number of accidents by relationship to junction.

12. Number of accidents involving wild animals by day of week.
13. Number of accidents involving domestic animals by day of week.
14. Number of accidents involving wild animals by month.
15. Number of accidents involving domestic animals by month.
16. Number of accidents involving wild animals by hour.
17. Number of accidents involving domestic animals by hour.
18. Number of accidents involving wild animals by light condition.
19. Number of accidents involving domestic animals by light condition.

2∅. Number of investigated accidents by hazardous moving violations.
21. Number of investigated accidents by traffic controls.
22. Number of investigated accidents by road defects.

23. Number of accidents involving contributing circumstances by contributing circumstances.

24. Number of vehicles by state in which registered.
25. Number of vehicles by body style.
26. Number of vehicles by trailer style.
27. Number of vehicles by damage severity.
28. Number of vehicles by vision.
29. Number of vehicles by mechanical defects.
3Ø. Number of vehicles by age.

31. Number of drivers by sex.
32. Number of drivers by age.
33. Number of drivers by physical defects.
34. Number of drivers by sobriety.
35. Number of drivers by re-examination recommendation.

36. Number of injuries by county.
37. Number of injuries by sex.
38. Number of injuries by location.
39. Number of injuries by sobriety.
4Ø. Number of injuries by age.
41. Number of injuries by severity.

Summaries 1 through 11 show number of selected accidents. Fatal accidents are shown separately.

Summaries 12 through 19 show only accidents that involved wild or domestic accidents. Fatal accidents are shown separately.

Summaries 2Ø through 22 show only investigated accidents. Fatal accidents are shown separately.

Summary 23 shows only those accidents that involved contributing circumstances. Fatal accidents are shown separately.

Summaries 24 through 3Ø show the number of vehicles involved in the accidents printed in summaries 1 through 11. Vehicles involved in fatal accidents are shown separately.

Summaries 31 through 35 show the number of drivers involved in the accidents shown in summaries 1 through 11. Drivers involved in fatal accidents are shown separately. The number of drivers may differ from the number of vehicles because drivers of hit-and-run vehicles are not included and because some of the vehicles may have been parked with no persons inside.

Summaries 36 through 41 show the number of injuries and fatalities involved in the accidents shown in summaries 1 through 11. Injuries and fatalities are shown separately.

A complete SUM-BY-TRAFFICWAY run consists of 2Ø printed pages. Any of the parameters shown above for FORM-16 commands may be used on SUM-BY-TRAFFICWAY commands.

Figures 2-1∅ through 2-15 on the following pages show a partial output from SUM-BY-TRAFFICWAY. Figure 2-1∅ shows summaries 8 through 11. Figure 2-11 shows summaries 17 and 18. Figure 2-12 shows summaries 2∅ through 22. Figure 2-13 shows summaries 26 through 28. Figure 2-14 shows summaries 31 through 34. Figure 2-15 shows part of summary 36 and summaries 37 through 39. These pages were obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:SUM-BY-TRAFFICWAY,START-DATE=∅7/∅1/74,END-DATE=∅7/31/74
/*
```

SUM-BY-TRAFFICWAY accesses the same files under the same conditions as shown above for FORM-16, except that the tables are always accessed.

<u>The MOTORCYCLE-SUMMARY Program</u>

MOTORCYCLE-SUMMARY prints a set of 21 summaries pertaining to motorcycle accidents. These summaries are:

1. Motorcycle accident totals.
2. Motorcyclists by injury severity.
3. Motorcycle accidents by type.
4. Motorcycle accidents by roadway location.
5. Motorcycle accidents by type of roadway.
6. Motorcycle accidents by city population.
7. Type of vehicle colliding with cycle in multi-vehicle accidents.
8. Motorcycle accident analysis.
9. Defects of motorcycles in investigated accidents.
1∅. Age of motorcyclist involved in accidents.
11. Sex of motorcyclists.
12. Motorcyclist violations in investigated accidents.
13. Other vehicle violations in investigated accidents.
14. Motorcyclist sobriety.
15. Time of accidents.
16. Accidents by day of week.
17. Accidents by month.
18. Accidents by light conditions.
19. Accidents by weather conditions.
2∅. Accidents by road conditions.
21. Accidents by county.

11 printed pages are generated by each MOTORCYCLE-SUMMARY command.

STATEWIDE ACCIDENTS   REPORTING PERIOD FROM 07/01/74 TO 07/31/74

ACCIDENT SUMMARIES.
UNDER "FATAL" IS NUMBER OF FATAL ACCIDENTS.
UNDER "TOTAL" IS NUMBER OF FATAL, INJURY AND PROPERTY DAMAGE ACCIDENTS.

	INTERSTATE		US HIGHWAY		STATE HIGHWAYS		COUNTY ROADS		LOCAL STREET		***TOTALS***	
	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL
ACCIDENTS BY WEATHER CONDITION												
CLEAR	2	84	15	517	5	291	1	217	3	338	26	1,447
RAINING		5	1	24	2	16		12		14	3	71
SNOWING						1						1
FOG												
OTHER		4	1	11		10		2		3	1	30
NOT STATED				7		1		4		4		16
TOTAL	2	93	17	559	7	319	1	235	3	359	30	1,565
ACCIDENTS BY ROAD CONDITION												
DRY	2	88	16	522	5	297	1	217	3	338	27	1,462
WET		5	1	30	2	19		12		17	3	83
SNOWY										1		1
ICY												
OTHER				1		2		3				6
NOT STATED				6		1		3		3		13
TOTAL	2	93	17	559	7	319	1	235	3	359	30	1,565
ACCIDENTS BY LIGHT CONDITION												
DAYLIGHT	2	62	11	359	5	217	1	159	1	243	20	1,040
DAWN OR DUSK		4		25		15		9		17		70
DARKNESS, LIT		3	1	61		21		3	2	74	3	162
DARKNESS, UNLIT		24	5	106	2	64		61		23	7	278
OTHER						1						1
NOT STATED				8		1		3		2		14
TOTAL	2	93	17	559	7	319	1	235	3	359	30	1,565
ACCIDENTS BY RELATIONSHIP TO JUNCTION												
NON-JUNCTION	2	83	8	304	7	184	1	192	3	162	21	925
INTERSECTION		5	5	111		66		24		133	5	339
INTERSECTION-RELATED		5	2	100		47		11		43	2	206
DRIVEWAY ACCESS			2	44		22		8		21	2	95
TOTAL	2	93	17	559	7	319	1	235	3	359	30	1,565

Figure 2-1Ø. SUM-BY-TRAFFICWAY Program.

STATEWIDE ACCIDENTS   REPORTING PERIOD FROM 07/01/74 TO 07/31/74

ACCIDENT SUMMARIES.
UNDER "FATAL" IS NUMBER OF FATAL ACCIDENTS.
UNDER "TOTAL" IS NUMBER OF FATAL, INJURY AND PROPERTY DAMAGE ACCIDENTS.

	INTERSTATE FATAL	INTERSTATE TOTAL	US HIGHWAY FATAL	US HIGHWAY TOTAL	STATE HIGHWAYS FATAL	STATE HIGHWAYS TOTAL	COUNTY ROADS FATAL	COUNTY ROADS TOTAL	LOCAL STREET FATAL	LOCAL STREET TOTAL	***TOTALS*** FATAL	***TOTALS*** TOTAL
ACCIDENTS INVOLVING DOMESTIC ANIMALS BY HOUR OF OCCURRENCE												
12:00 M - 12:59 AM				1				1				2
1:00 - 1:59				2								2
2:00 - 2:59		1										1
3:00 - 3:59		1						1				2
4:00 - 4:59												
5:00 - 5:59				1		1						2
6:00 - 6:59												
7:00 - 7:59						2		1				3
9:00 - 9:59						1		1				2
8:00 - 8:59												
10:00 - 10:59												
11:00 - 11:59		1		1				2				4
12:00 N - 12:59 PM								1		1		2
1:00 - 1:59												
2:00 - 2:59								1				1
3:00 - 3:59						1				1		2
4:00 - 4:59				1								1
5:00 - 5:59										1		1
6:00 - 6:59			1	1							1	1
7:00 - 7:59												
8:00 - 8:59						1						1
9:00 - 9:59				1				1		1		3
10:00 - 10:59				8		1		2		1		12
11:00 - 11:59		1		6		1				2		10
NOT STATED												
TOTAL		4	1	22		8		11		7	1	52
ACCIDENTS INVOLVING WILD ANIMALS BY LIGHT CONDITION												
DAYLIGHT		1		2		4		1				8
DAWN OR DUSK				3		2		1				6
DARKNESS, LIT												
DARKNESS, UNLIT		4		9		4		1				18
OTHER												
NOT STATED				2								2
TOTAL		5		16		10		3				34

Figure 2-11. SUM-BY-TRAFFICWAY Program.

STATEWIDE ACCIDENTS   REPORTING PERIOD FROM 07/01/74 TO 07/31/74

ACCIDENT SUMMARIES.
UNDER "FATAL" IS NUMBER OF FATAL ACCIDENTS.
UNDER "TOTAL" IS NUMBER OF FATAL, INJURY AND PROPERTY DAMAGE ACCIDENTS.

	INTERSTATE FATAL	INTERSTATE TOTAL	US HIGHWAY FATAL	US HIGHWAY TOTAL	STATE HIGHWAYS FATAL	STATE HIGHWAYS TOTAL	COUNTY ROADS FATAL	COUNTY ROADS TOTAL	LOCAL STREET FATAL	LOCAL STREET TOTAL	***TOTALS*** FATAL	***TOTALS*** TOTAL
INVESTIGATED ACCIDENTS BY HAZARDOUS MOVING VIOLATIONS												
ACCIDENTS INVOLVING HMV	1	34	1	239		163		109		121	2	671
ACCIDENTS NOT INVOLVING HMV	1	49	16	206	7	113	1	77	3	106	28	553
TOTAL	2	83	17	445	7	281	1	186	3	229	30	1,224
INVESTIGATED ACCIDENTS BY TRAFFIC CONTROLS												
TRAFFIC SIGNALS				52		21				14		87
TRAFFIC SIGNALS NOT WORKING				1		2		2				5
TRAFFIC SIGNALS W/PED HEADS				4						3		7
TRAFFIC SIGNALS W/HEADS NOT WORKING										1		1
FLASHER				4		3				1		8
FLASHER NOT WORKING				1								1
STOP SIGN		6	2	25		20		9		22	2	83
YIELD SIGN				1		2		1		4		8
RAILROAD SIGNALS				1		1						2
RAILROAD SIGNALS NOT WORKING				2								2
RAILROAD GATES				1						1		2
RAILROAD GATES NOT WORKING				2								2
DO NOT ENTER SIGNS				2								2
OTHER REGULATORY SIGN		7	1	36	2	30		27		20	3	120
WARNING SIGN		5		14		4		6	1	2	1	31
PAVEMENT MARKINGS		15	5	82	2	45		2	1	13	8	157
NOT STATED/NO SIGNALS	2	50	9	216	3	153	1	139	1	148	16	706
TOTAL	2	83	17	445	7	281	1	186	3	229	30	1,224
INVESTIGATED ACCIDENTS BY ROAD DEFECTS												
HOLES OR RUTS						2		6				8
SHOULDER						2		2				4
LOOSE MATERIAL				1		9		34	1	12	1	56
CONSTRUCTION		2		4				1				7
OTHER		2		2		2		4				10
NO ROAD DEFECTS	2	79	17	438	7	266	1	139	2	217	29	1,139
TOTAL	2	83	17	445	7	281	1	186	3	229	30	1,224

Figure 2-12. SUM-BY-TRAFFICWAY Program.

STATEWIDE ACCIDENTS   REPORTING PERIOD FROM 07/01/74 TO 07/31/74

VEHICLE SUMMARIES.
UNDER "FATAL" IS NUMBER OF VEHICLES INVOLVED IN FATAL ACCIDENTS.
UNDER "TOTAL" IS NUMBER OF VEHICLES INVOLVED IN ALL ACCIDENTS.

	INTERSTATE		US HIGHWAY		STATE HIGHWAYS		COUNTY ROADS		LOCAL STREET		***TOTALS***	
	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL
VEHICLES BY TRAILER STYLE												
CAMPING TRAILER		5	2	13		3					2	21
MOBILE HOME		2		5								7
UTILITY TRAILER		4		6		1		2		1		14
BOAT TRAILER				2		1						3
SEMI TRAILER		4	2	16		7		2			2	29
COMM. CARGO TRAILER		3		9		3						15
OTHER TRAILER		3		6		5		1				15
NO TRAILER	2	93	24	877	9	481	1	312	3	684	39	2,447
TOTAL	2	114	28	934	9	501	1	317	3	685	43	2,551
VEHICLES BY DAMAGE SEVERITY												
DISABLING DAMAGE	2	67	23	320	7	208	1	152	3	179	36	926
FUNCTIONAL DAMAGE		22	4	355	1	161		101		294	5	933
OTHER MOTOR VEHICLE DAMAGE		21	1	136	1	150		52		157	2	516
NO DAMAGE		4		73		32		12		55		176
TOTAL	2	114	28	934	9	501	1	317	3	685	43	2,551
VEHICLES BY VISION												
BUILDINGS								2		6		8
TREES OR HEDGES			1	4		3		10		10	1	27
OTHER VEHICLE		3		11		4		3		7		28
SMOKE												
DUST				2		2		5		1		10
OTHER		4	1	18	3	11		14		11	4	58
VISION NOT OBSCURED	2	107	26	859	6	481	1	283	3	650	38	2,420
TOTAL	2	114	28	934	9	501	1	317	3	685	43	2,551

Figure 2-13. SUM-BY-TRAFFICWAY Program.

STATEWIDE ACCIDENTS REPORTING PERIOD FROM 07/01/74 TO 07/31/74

DRIVER SUMMARIES.
UNDER "FATAL" IS NUMBER OF DRIVERS INVOLVED IN FATAL ACCIDENTS.
UNDER "TOTAL" IS NUMBER OF DRIVERS INVOLVED IN ALL ACCIDENTS.

	INTERSTATE		US HIGHWAY		STATE HIGHWAYS		COUNTY ROADS		LOCAL STREET		***TOTALS***	
	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL	FATAL	TOTAL
DRIVERS BY SEX												
MALE		86	24	670	6	334	1	250	3	381	34	1,721
FEMALE	2	24	4	220	3	143		47		176	9	610
NOT STATED						1						1
TOTAL	2	110	28	890	9	478	1	297	3	557	43	2,332
DRIVERS BY AGE												
15 YEARS AND YOUNGER		2	1	19		17		14		16	1	68
16 YEARS		2		38	2	16		29	1	37	3	122
17 YEARS		6	3	43	1	33		20		49	4	151
18 TO 19 YEARS		10	4	112	1	62		49	1	89	6	322
20 TO 24 YEARS		21	4	154	3	87		61	1	100	8	423
25 TO 34 YEARS		28	5	174		86	1	51		76	6	415
35 TO 44 YEARS	1	12	3	106	1	70		25		39	5	252
45 TO 54 YEARS	1	15	2	90	1	33		15		45	4	198
55 TO 64 YEARS		7	2	70		30		16		48	2	171
65 TO 74 YEARS		4	4	49		19		6		20	4	98
75 YEARS AND OLDER		1		8		11		3		12		35
NOT STATED		2		27		14		8		26		77
TOTAL	2	110	28	890	9	478	1	297	3	557	43	2,332
DRIVERS BY PHYSICAL DEFECTS												
VISION		1		5		3				3		12
HEARING				1						1		2
ILLNESS		1	1	1		2					1	4
MISSING LIMBS				1						2		3
OTHER			1	5		6		6		3	1	20
NO APPARENT DEFECTS	2	108	26	677	9	467	1	291	3	548	41	2,291
TOTAL	2	110	28	890	9	478	1	297	3	557	43	2,332
DRIVERS BY SOBRIETY												
HAD BEEN DRINKING	1	16	6	117	3	75	1	58	2	46	13	312
HAD NOT BEEN DRINKING	1	94	22	773	6	403		239	1	511	30	2,020
TOTAL	2	110	28	890	9	478	1	297	3	557	43	2,332

Figure 2-14. SUM-BY-TRAFFICWAY Program.

STATEWIDE ACCIDENTS REPORTING PERIOD FROM 07/01/74 TO 07/31/74

INJURY SUMMARIES.
UNDER "FATAL" IS NUMBER OF FATALITIES.
UNDER "INJURY" IS NUMBER OF INJURIES (FATALITIES ARE NOT INCLUDED).

	INTERSTATE FATAL	INTERSTATE INJURY	US HIGHWAY FATAL	US HIGHWAY INJURY	STATE HIGHWAYS FATAL	STATE HIGHWAYS INJURY	COUNTY ROADS FATAL	COUNTY ROADS INJURY	LOCAL STREET FATAL	LOCAL STREET INJURY	***TOTALS*** FATAL	***TOTALS*** INJURY
TETON				6		1		6		3		16
TOOLE		1							1	3	1	4
TREASURE												
VALLEY				10				1		1		12
WHEATLAND				1								1
WIBAUX		2										2
YELLOWSTONE		6		23		42		15		17		103
TOTAL	2	73	20	337	8	209	1	157	3	143	34	919
INJURIES BY SEX												
MALE	1	44	11	192	6	140	1	121	2	79	21	576
FEMALE	1	29	9	145	2	68		36	1	64	13	342
NOT STATED						1						1
TOTAL	2	73	20	337	8	209	1	157	3	143	34	919
INJURIES BY LOCATION												
DRIVER		36	12	171	5	118	1	90	1	70	19	485
PASSENGER	2	35	6	153	2	84		64	1	57	11	393
PEDESTRIAN		2	1	4	1	2		1	1	9	3	18
BICYCLIST			1	9		5		2		7	1	23
TOTAL	2	73	20	337	8	209	1	157	3	143	34	919
INJURIES BY SOBRIETY												
HAD BEEN DRINKING	1	12	5	56	3	62	1	37	1	24	11	191
HAD NOT BEEN DRINKING	1	61	15	281	5	147		120	2	119	23	728
TOTAL	2	73	20	337	8	209	1	157	3	143	34	919

Figure 2-15. SUM-BY-TRAFFICWAY Program.

To use the MOTORCYCLE-SUMMARY program, prepare a command that specifies the MOTORCYCLE-SUMMARY program. Any of the parameters shown above for FORM-16 can be included on the command.

Figure 2-16 on the following pages show a partial output from a MOTORCYCLE-SUMMARY run. The output was produced with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:MOTORCYCLE-SUMMARY,START-DATE=Ø7/Ø1/74,END-DATE=Ø7/31/74
/*
```

MOTORCYCLE-SUMMARY accesses the same files under the same conditions as shown above for FORM-16, except that the tables are always accessed.

The RURAL-ACC-CLUSTERS Program

RURAL-ACC-CLUSTERS is one of several programs available for rural accident studies. It can be used for locating "clusters" of accidents along rural highways. The RURAL-ACC-ANALYSIS program can be used for more in-depth analysis of clusters after they have been located. The Department of Highways has programmed two additional rural accident analysis reports - one for analyzing rural intersections and one for analyzing open range areas. Information on these programs can be obtained from the Data Processing Bureau.

The RURAL-ACC-CLUSTERS program operates by searching rural highways for sections of a user-specified length on which a user-specified number of accidents occurred. Only federal-aid highways can be searched. Each time a cluster is located, a listing of the accidents in the section is printed.

The program uses a "sliding" section to search for clusters. At each accident, the program looks ahead the specified section length and counts the number of accidents in the section. If the count equals or exceeds the specified count, the cluster is printed. The program then skips to the next accident and looks ahead from it. A given accident can therefore appear in several clusters.

When a cluster contains more accidents than the user-specified count, it may contain several "sub-clusters." For example, suppose the trigger level is 15 accidents and a cluster is found that contains 2Ø accidents. When the program prints the cluster and then skips to the second accident of the cluster, a cluster

MOTORCYCLE SUMMARY PARAMETERS
START-DATE=07/01/74 END-DATE=07/31/74 LOCATION

MOTORCYCLE ACCIDENT TOTALS
....

TOTAL ACCIDENTS....................	108
FATAL ACCIDENTS....................	5
INJURY ACCIDENTS...................	93
PERSONS KILLED.....................	6
PERSONS INJURED....................	119

MOTORCYCLISTS	NUMBER	% OF TOTAL
FATAL	6	4
INJURED	114	87
SEVERITY OF INJURY:		
INCAPACITATING	44	33
NON-INCAPACITATING	53	40
POSSIBLE INJURY	17	12
ESCAPING INJURY	11	8
TOTAL PERSONS	131	100

MOTORCYCLE ACCIDENTS BY TYPE

TYPE OF ACCIDENT	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
NOT STATED						
NON-COLLISION:						
OVERTURNING	29	1	2		27	1
OTHER NON-COLLISION						
COLLISION WITH:						
PEDESTRIAN						
MOTOR VEHICLE IN TRANSPORT	64	3	31		33	3
VEHICLE IN OTHER ROADWAY						
PARKED MOTOR VEHICLE	1				1	
RAILWAY TRAIN						
PEDALCYCLIST	1				1	
ANIMAL	2		1		1	
FIXED OBJECT	11	1	3		8	1
OTHER OBJECT						
T O T A L	108	5	37		71	5

Figure 2-16. MOTORCYCLE-SUMMARY Program. Page 1 of 5.

MOTORCYCLE ACCIDENT LOCATION

ACCIDENT LOCATION	URBAN AREAS TOTAL	URBAN AREAS % OF TOTAL	URBAN AREAS FATAL	RURAL AREAS TOTAL	RURAL AREAS % OF TOTAL	RURAL AREAS FATAL
ON ROADWAY	33	89		49	69	3
OFF ROADWAY	4	10		22	30	2
T O T A L	37	100		71	100	5

TYPE OF ROADWAY	TOTAL	% OF TOTAL	FATAL	TOTAL	% OF TOTAL	FATAL
INTERSTATE SYSTEM				4	5	
U. S. NO. ROUTE	15	40		24	33	3
STATE NO. ROUTE	5	13		13	18	1
COUNTY ROAD				24	33	
CITY STREET	17	45		2	2	
OTHER TRAFFICWAYS				4	5	1
T O T A L	37	100		71	100	5

CITIES OVER 2,500	TOTAL	% OF TOTAL	FATAL
2500 - 5000	3	8	
5000 - 10,000	2	5	
10,000 - 25,000	6	16	
25,000 - 50,000	9	24	
50,000 OR MORE	17	45	
T O T A L	37	100	

TYPE OF VEHICLE COLLIDING WITH CYCLE IN MULTI-VEHICLE CYCLE ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
PASSENGER CAR	42	2	22		20	2
PICKUP	13	1	4		9	1
TRUCK	2		1		1	
BUS OR SCHOOL BUS						
OTHER MOTORCYCLE	2				2	
OTHER	7		4		3	
T O T A L	66	3	31		35	3

Figure 2-16. MOTORCYCLE-SUMMARY Program. Page 2 of 5.

MOTORCYCLE ACCIDENT ANALYSIS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
SINGLE VEHICLE ACCIDENTS:						
NOT STATED						
NON-COLLISION:						
OVERTURNED	29	1	2		27	1
OTHER NON-COLLISION						
COLLISION WITH:						
PEDESTRIAN						
TRAIN						
PEDALCYCLIST						
ANIMAL	2		1		1	
FIXED OBJECT	11	1	3		8	1
OTHER OBJECT						
SUB-TOTAL	42	2	6		36	2
MULTIVEHICLE ACCIDENTS:						
AT INTERSECTION:						
HEAD-ON						
REAR-END						
ENTERING AT AN ANGLE	18	1	12		6	1
ONE VEHICLE TURNING	12	1	8		4	1
OTHER						
SUB-TOTAL	30	2	20		10	2
NOT AT AN INTERSECTION:						
MOVING OPPOSITE DIRECTIONS	8		1		7	
MOVING SAME DIRECTION	6		2		4	
ONE VEHICLE PARKED	21	1	7		14	1
ONE VEHICLE STOPPED IN TRAFFIC	1		1			
ONE VEHICLE LEAVING PARKED POS						
LEAVING/ENTERING DRIVEWAY						
OTHERS						
SUB-TOTAL	36	1	11		25	1
TOTAL	108	5	37		71	5

Figure 2-16. MOTORCYCLE-SUMMARY Program. Page 3 of 5.

DEFECTS OF MOTORCYCLES IN INVESTIGATED TRAFFIC ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
LIGHTS	1				1	
BRAKES	1				1	
TIRES/STEERING	3				3	
OTHER	2				2	
TOTAL DEFECTS	7				7	
MOTORCYCLES WITH NO DEFECTS	87	5	31		56	5
T O T A L	94	5	31		63	5

AGE OF MOTORCYCLIST INVOLVED IN TRAFFIC ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
15 AND UNDER	10		2		8	
16 TO 19	51	2	20		31	2
20 TO 24	30	1	12		18	1
25 TO 34	26	2	7		19	2
35 TO 64	11	1	3		8	1
65 AND OVER						
NOT STATED	3		1		2	
T O T A L	131	6	45		86	6

SEX OF MOTORCYCLE DRIVER INVOLVED IN MOTORCYCLE ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
MALE	107	5	36		71	5
FEMALE	2		1		1	
NOT STATED	1				1	
T O T A L	110	5	37		73	5

MOTORCYCLIST SOBRIETY IN MOTOR CYCLE ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
HAD BEEN DRINKING	17	3	4		13	3
HAD NOT BEEN DRINKING	114	3	41		73	3
T O T A L	131	6	45		86	6

Figure 2-16. MOTORCYCLE-SUMMARY Program. Page 4 of 5.

MOTORCYCLIST VIOLATIONS IN INVESTIGATED CYCLE ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
UNDER INFLUENCE OF ALCHOHOL						
UNDER INFLUENCE OF DRUGS						
EXCEEDING STATED SPEED LIMIT						
EXCEEDING SAFE SPEED	14		3		11	
DID NOT GRANT R. O. W. TO VEHICLE	4		2		2	
DID NOT GRANT R. O. W. TO PEDESTRIAN						
IMPROPER PASSING						
FOLLOWING TOO CLOSELY						
OVER CENTER LINE						
IMPROPER TURNING						
IMPROPER SIGNAL						
DISREGARDED TRAFFIC SIGNAL						
OPERATING DEFECTIVE EQUIPMENT						
INATTENTION	2				2	
OTHER	7		1		6	
T O T A L	27		6		21	
NUMBER OF INVESTIGATED ACCIDENTS	93	5	31		62	5

OTHER VEHICLE VIOLATIONS IN INVESTIGATED CYCLE ACCIDENTS	STATEWIDE TOTAL	STATEWIDE FATAL	URBAN TOTAL	URBAN FATAL	RURAL TOTAL	RURAL FATAL
UNDER INFLUENCE OF ALCHOHOL	1		1			
UNDER INFLUENCE OF DRUGS						
EXCEEDING STATED SPEED LIMIT						
EXCEEDING SAFE SPEED	2		2			
DID NOT GRANT R. O. W. TO VEHICLE	11		6		5	
DID NOT GRANT R. O. W. TO PEDESTRIAN						
IMPROPER PASSING						
FOLLOWING TOO CLOSELY						
OVER CENTER LINE						
IMPROPER TURNING	1				1	
IMPROPER SIGNAL						
DISREGARDED TRAFFIC SIGNAL						
OPERATING DEFECTIVE EQUIPMENT						
INATTENTION	1		1			
OTHER	3		2		1	
T O T A L	19		12		7	

Figure 2-16. MOTORCYCLE-SUMMARY Program. Page 5 of 5.

of 19 accidents will be found. These sub-clusters are not printed unless at least one more accident is added to the cluster. Hence, if this second cluster contains only the 19 accidents that have already been printed, it is not printed. If one or more additional accidents appear in the cluster, it is printed (the 19 that have already been printed are flagged as being previously printed in the print-out of the second cluster). The user can specify that these second clusters be printed only if a specified number of additional accidents are found. For example, the trigger level can be specified as 15 accidents but that clusters be printed only if they contain at least 5 accidents that have not already been printed.

To use the clusters program, prepare a command that specifies the RURAL-ACC-CLUSTERS program. Specify the length in miles of the sliding section by coding LENGTH=nn.nn. The parameter must be coded in exactly this format. For example, to specify a length of 3/4 mile, code LENGTH=ØØ.75. To specify a length of one mile, code LENGTH=Ø1.ØØ.

Specify the trigger level by coding #-ACCIDENTS=n (code a value ranging from 1 through 999). If desired, specify a secondary trigger level by coding #-NEW-ACC=n (code a value ranging from 1 through 999 but less than or equal to the value in #-ACCIDENTS). For the above example using a trigger level of 15 accidents and a secondary trigger level of 5 accidents, code #-ACCIDENTS=15,#-NEW-ACC=5.

A DATA parameter (see chapter 1) must be included on the command to identify which highway(s) are to be scanned. Only federal aid highways can be specified in the DATA parameter.

The following selection parameters can be included on the command (these parameters are described earlier in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
LOCATION or COUNTY
START-MILEPOINT and END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and SELECT-SIZE

Any of the print parameters described in chapter 1 can be coded on the command.

Figure 2-17 on the following pages shows a sample RURAL-ACC-CLUSTERS output. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC H1SACCA
//SYSIN DD *
:RURAL-ACC-CLUSTERS,LENGTH=Ø1.ØØ,#-ACCIDENTS=24,
:   START-DATE=Ø1/Ø1/72,END-DATE=12/31/74,
:   DATA=PRIMARY=8
/*
```

RURAL-ACC-CLUSTERS accesses the following files:

DD Statement	File
ACIDENT	Accident detail file - Accessed for LOCATION=STATEWIDE, COUNTY=county, or a select statement that refers to the accident detail file.
ACCVEH	Accident vehicle file - Accessed if a select statement refers to the accident vehicle file.
ACCDIRI	Accident directory file - Always accessed.
ROADLOG	Roadlog file - Accessed if a select statement refers to the roadlog file.
TRAFFIC	Traffic file - Always accessed (used to locate spurs, loops, and coincident sections).
TRUMILE	True mileage file - Always accessed.
TABLES	Tables - Accessed if COUNTY or SELECT-DD specified.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

The following error messages can be printed by RURAL-ACC-CLUSTERS:

***** PROGRAM ERROR - INCREASE SPUR/LOOP ARRAY *****
***** PROGRAM ERROR - INCREASE COINCIDENT ARRAY *****

These messages indicate a program error that must be corrected by system maintenance personnel. Refer the problem to the Data Processing Bureau.

***** ACCIDENT /accident-number/ AT /location/ NOT USED - NOT SPECIFIED TO NEAREST REFERENCE POST

An accident was skipped due to a data error.

OCT 29,1975

ROUTE NUMBER	MILEPOINT	ACCIDENT NUMBER	NO. FAT.	NO. INJ.	NO. LANES	DATE	HOUR	FIRST-HARMFUL-EVENT	TYPE-OF-COLLISION	ROAD SURFACE	DIST

***** ACCIDENT /730002110499/ AT /P00008034+0.980/ NOT USED - NOT SPECIFIED TO NEAREST REFERENCE POST
- - - - - - - -KEY SHOULD BE P00008035+0.053

ROUTE NUMBER	MILEPOINT	ACCIDENT NUMBER	NO. FAT.	NO. INJ.	NO. LANES	DATE	HOUR	FIRST-HARMFUL-EVENT	TYPE-OF-COLLISION	ROAD SURFACE	DIST
P00008	045+0.450	720001430902			4	9/29/72	16	MV IN TRANSPORT	ANGLE	DRY	0.000
P00008	045+0.460	730001920401			4	4/09/73	16	MV IN TRANSPORT	SIDESWIPE--PASSING	DRY	0.010
P00008	045+0.460	730002881203			4	12/22/73	19	FIXED OBJECT		WET	0.010
P00008	045+0.460	740002600905			4	9/15/74	06	MV IN TRANSPORT	REAR END	DRY	0.010
P00008	045+0.460	740002601101		3	4	11/06/74	11	MV IN TRANSPORT	ANGLE	DRY	0.010
P00008	045+0.460	740002881005			4	10/21/74	16	MV IN TRANSPORT	ANGLE	WET	0.010
P00008	045+0.600	730002760803			4	8/05/73	00	FIXED OBJECT		DRY	0.150
P00008	045+0.800	740002600602	1		4	6/01/74	22	PEDESTRIAN		DRY	0.350
P00008	045+0.800	740002600603		1	4	6/01/74	22	MV IN TRANSPORT	REAR END	DRY	0.350
P00008	045+0.800	740002760801			4	8/02/74	21	MV IN TRANSPORT	SIDESWIPE--PASSING	DRY	0.350
P00008	045+0.960	740001410201		1	4	2/14/74	07	MV IN TRANSPORT	REAR END	DRY	0.510
P00008	046+0.000	740002761201			4	12/03/74	15	MV IN TRANSPORT		DRY	0.578
P00008	046+0.100	720002111204			2	12/17/72	13	MV IN TRANSPORT	REAR END	DRY	0.678
P00008	046+0.100	730002600502		2	2	5/05/73	21	MV IN TRANSPORT	REAR END	DRY	0.678
P00008	046+0.100	730002601107		2	2	11/18/73	16	PEDESTRIAN		ICY	0.678
P00008	046+0.100	730002761207		1	2	12/18/73	00	MV IN TRANSPORT	SIDESWIPE--PASSING	ICY	0.678
P00008	046+0.200	720001850609		1	2	6/29/72	22	MV IN TRANSPORT	ANGLE	DRY	0.778
P00008	046+0.200	720002110802			2	8/09/72	01	MV IN TRANSPORT	REAR END	DRY	0.778
P00008	046+0.200	740001241002			2	10/10/74	23	MV IN TRANSPORT	REAR END	DRY	0.778
P00008	046+0.250	730005104081			2	10/10/73	00	OVERTURNING	SIDESWIPE--PASSING	DRY	0.828
P00008	046+0.300	730001850801		1	2	8/13/73	23	MV IN TRANSPORT	REAR END	DRY	0.878
P00008	046+0.300	730002600505			2	5/22/73	23	MV IN TRANSPORT	ANGLE	DRY	0.878
P00008	046+0.300	740001850805		6	2	8/29/74	15	MV IN TRANSPORT	HEAD ON	DRY	0.878
P00008	046+0.400	720002890201		2	2	2/13/72	17	OVERTURNING		ICY	0.978

-----> THERE WERE 24 ACCIDENTS IN THIS SECTION

ROUTE NUMBER	MILEPOINT	ACCIDENT NUMBER	NO. FAT.	NO. INJ.	NO. LANES	DATE	HOUR	FIRST-HARMFUL-EVENT	TYPE-OF-COLLISION	ROAD SURFACE	DIST
P00008	047+0.500	740002881001			2	10/06/74	01	OVERTURNING		DRY	0.000
P00008	047+0.560	72P005C5A001			2	5/04/72	16	MV IN TRANSPORT	ANGLE	DRY	0.060
P00008	047+0.600	730002111003		2	2	10/15/73	08	MV IN TRANSPORT	ANGLE	DRY	0.100
P00008	047+0.760	720002761005			2	10/10/72	23	FIXED OBJECT		ICY	0.260
P00008	047+0.770	740001411105			2	11/27/74	18	FIXED OBJECT		ICY	0.270
P00008	047+0.770	740001411106		1	2	11/27/74	18	MV IN TRANSPORT	HEAD ON	ICY	0.270
P00008	047+0.800	740002600902		1	2	9/05/74	21	MV IN TRANSPORT	SIDESWIPE--MEETING	DRY	0.300
P00008	047+0.860	730001420301			2	3/02/73	17	MV IN TRANSPORT	SIDESWIPE--PASSING	DRY	0.360
P00008	047+0.900	730001850606		4	2	6/30/73	07	FIXED OBJECT		DRY	0.400
P00008	047+0.900	740002760503			2	5/30/74	22	MV IN TRANSPORT	REAR END	WET	0.400
P00008	047+0.900	740002880803			2	8/19/74	10	MV IN TRANSPORT	SIDESWIPE--MEETING	DRY	0.400
P00008	047+0.900	740002970201			2	2/27/74	19	MV IN TRANSPORT	REAR END	DRY	0.400
P00008	047+0.960	720001850705		1	2	7/24/72	14	MV IN TRANSPORT	SIDESWIPE--PASSING	DRY	0.460
P00008	047+0.960	730002760104			2	1/24/73	09	MV IN TRANSPORT	REAR END	DRY	0.460
P00008	048+0.000	720002951001		1	2	10/28/72	15	PARKED MV	REAR END	ICY	0.484
P00008	048+0.000	740001850202			2	2/06/74	19	MV IN TRANSPORT	REAR END	ICY	0.484
P00008	048+0.070	730002111202			2	12/31/73	22	MV IN TRANSPORT	SIDESWIPE--MEETING	SNOWY	0.554
P00008	048+0.100	720002760507			2	5/24/72	20	FIXED OBJECT		DRY	0.584
P00008	048+0.200	730002880503			2	5/06/73	18	OTHER NONCOLLISION		DRY	0.684
P00008	048+0.200	740002880805			2	8/25/74	02	MV IN TRANSPORT	REAR END	DRY	0.684

Figure 2-17. RURAL-ACC-CLUSTERS Program. Page 1 of 2.

OCT 29,1975

ROUTE NUMBER	MILEPOINT	ACCIDENT NUMBER	NO. FAT.	NO. INJ.	NO. LANES	DATE	HOUR	FIRST-HARMFUL-EVENT	TYPE-OF-COLLISION	ROAD SURFACE	DIST
P00008	048+0.360	720002760902	1	1	2	9/03/72	00	OVERTURNING		DRY	0.844
P00008	048+0.420	748005114003		1	2	11/27/74	00	FIXED OBJECT		ICY	0.904
P00008	048+0.460	730001920301			2	3/09/73	17	MV IN TRANSPORT	REAR END	DRY	0.944
P00008	048+0.500	720001850302			2	3/07/72	18	MV IN TRANSPORT	REAR END	DRY	0.984
P00008	048+0.500	740002600102		1	2	1/11/74	23	OTHER OBJECT	REAR END	DRY	0.994
P00008	048+0.500	740002601107	2	3	2	11/28/74	01	MV IN TRANSPORT	HEAD ON	DRY	0.934

-----> THERE WERE 26 ACCIDENTS IN THIS SECTION

***** ACCIDENT /720002111203/ AT /P00008050+1.400/ NOT USED - NOT SPECIFIED TO NEAREST REFERENCE POST
- - - - - - -KEY SHOULD BE P00008051+0.409

-----> THERE WERE 2 SECTIONS WITH AT LEAST 24 ACCIDENTS WITHIN A DISTANCE OF 01.00 MILES.

Figure 2-17. RURAL-ACC-CLUSTERS Program. Page 2 of 2.

***** ACCIDENT /accident-number/ AT /location/ NOT USED - NO TRUE MILEAGE

An accident was skipped due to a data error.

***** ACCIDENT /accident-number/ AT /location/ NOT USED - ERROR IN KEY FORMAT

An accident was skipped due to a data error.

The RURAL-ACC-ANALYSIS Program

RURAL-ACC-ANALYSIS prints a detailed accident summary for a stretch of highway. Unlike the RURAL-ACC-CLUSTERS program which can locate clusters of accidents, RURAL-ACC-ANALYSIS can be used only when a particular stretch of highway has already been chosen for analysis.

RURAL-ACC-ANALYSIS can be used only for federal aid routes. Each command can specify only one segment of highway which can be one route or a portion of a route. The segment cannot contain any discontinuities such as spurs, loops, or coincident sections (coincident sections can only by analyzed for the "base" route).

Each RURAL-ACC-ANALYSIS command prints the following outputs:

1. A straight-line map of the roadway scaled to fit one printed page and showing accidents and roadway descriptions.
2. A summary page showing traffic data and total accident counts.
3. A listing of the accidents showing date and location information.
4. A listing of the accidents showing accident details.
5. A listing of the vehicle and pedestrian information for the accidents.

To use the analysis program, prepare a command that specifies the RURAL-ACC-ANALYSIS program. Include a DATA parameter, a START-MILEPOINT parameter, and an END-MILEPOINT parameter (these parameters are described in chapter 1). The DATA parameter must specify a single federal aid route (eg., DATA=INTERSTATE=9Ø or DATA=PRIMARY=8). Be sure that no discontinuities in the highway occur between the starting and ending milepoints coded. If desired, the following parameters can be included on the command (these parameters are described in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE

Any of the print parameters described in chapter 1 can be included on the command.

Figure 2-18 on the following pages show a sample RURAL-ACC-ANALYSIS run. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:RURAL-ACC-ANALYSIS,DATA=PRIMARY=8,
:  START-MILEPOINT=Ø26+Ø.ØØØ,END-MILEPOINT=Ø28+Ø.ØØØ,
:  START-DATE=Ø1/Ø1/72,END-DATE=12/31/74
/*
```

RURAL-ACC-ANALYSIS accesses the following files:

DD Statement	File
ACIDENT	Accident detail file - Always accessed.
ACCDIRI	Accident directory file - Always accessed.
ACCVEH	Accident vehicle file - Always accessed.
TRUMILE	True mileage file - Always accessed.
ROADLOG	Roadlog file - Always accessed.
TRAFFIC	Traffic file - Always accessed.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

```
RURAL ACCIDENT ANALYSIS OF THE  ACCIDENTS OCCURRING BETWEEN MILEPOINT P00008026+0.000 & P00008028+0.000

     _<---025+0.581 BEG 4LNU POWELL CO                   BUILT 1973    4 LANES, 14 FT. LANE, UNDIVIDED
     .                                                      FT. SHOULDER, MIXED BITUMINOUS ROAD SURFACE,
     . 0.419 MILES
     .
ACCIDENT ANALYSIS BEGINS AT MILEPOST 026+0.000
   SCALE =   0.042 MILES PER LINE
     |
     |
     |
    X|
    X|
     |
    X|
     |
     |
     |
     |
     |
     |
     |
    0|<---026+0.677 ENT HELENA OF POWELL CO              BUILT 1973,    4 LANES, 14 FT. LANE, UNDIVIDED
     |                                                      FT. SHOULDERS, MIXED BITUMINOUS ROAD SURFACE,
     |
     |
     |
     |
     |
     |
    X|
     |
    X|<---027+0.157   POWELL COUNTY                      BUILT 1973,    4 LANES, 14 FT. LANE, UNDIVIDED
     |                                                      FT. SHOULDERS, MIXED BITUMINOUS ROAD SURFACE,
    0|
     |
     |<---027+0.313 END 4LN POWELL COUNTY                BUILT 1931,    2 LANES, 12 FT. LANE, UNDIVIDED
     |                                                    3 FT. SHOULDERS, MIXED BITUMINOUS ROAD SURFACE,
     |
    0|
     |
   00|
 00X0|<---027+0.617 POWELL CO LN LEWIS & CLARK CO        BUILT 1931,    2 LANES, 12 FT. LANE, UNDIVIDED
     |                                                    3 FT. SHOULDERS, MIXED BITUMINOUS ROAD SURFACE,
    0|
     |
    0|
     |
     |
     |
     |
    0|
MILEPOST 028+0.000

F = FATALITY  X = INJURY  O = PROPERTY DAMAGE ONLY
```

Figure 2-18. RURAL-ACC-ANALYSIS Program. Page 1 of 4.

```
**** THE AVERAGE DIALY TRAFFIC (ADT) BETWEEN MILEPOINTS P00008026+0.000 & P00008028+0.000 ****

     DURING 1972 THE AVERAGE DAILY TRAFFIC (ADT) ON THIS ACCIDENT SECTION WAS   1390
     DURING 1973 THE AVERAGE DAILY TRAFFIC (ADT) ON THIS ACCIDENT SECTION WAS   1476
     DURING 1974 THE AVERAGE DAILY TRAFFIC (ADT) ON THIS ACCIDENT SECTION WAS   1263


      THE WEIGHTED ANNUAL AVERAGE DAILY TRAFFIC(ADT) FOR THE ACCIDENT SECTION IS   1376

      THE AVERAGE VEHICLE MILEAGE FOR THE ACCIDENT SECTION IS      2047

      THE ACCIDENT RATE BASED ON THE AVERAGE ADT AND THE NUMBER ACCIDENTS OCCURRING ON
           THE ACCIDENT SECTION BETWEEN P00008026+0.000 & P00008028+0.000 IS   5.86
      THE NUMBER OF ACCIDENTS IN THIS SECTION IS  17
      THE NUMBER OF FATAL ACCIDENTS IN THIS SECTION IS   0
      THE NUMBER OF FATALITIES IN THIS SECTION IS   0
      THE NUMBER OF INJURY ACCIDENTS IN THIS SECTION IS   6
      THE NUMBER OF INJURIES IN THIS SECTION IS  10
     THE ACCIDENT SEVERITY FOR THIS SECTION IS ***  1.352 ***
```

Figure 2-18. RURAL-ACC-ANALYSIS Program. Page 2 of 4.

***** ACCIDENT LOCATIONS *****

REFERENCE POST	ACCIDENT NO.	TIME	DATE	DAY OF WEEK	CLASS OF TRAFFICWAY	JUNCTION RELATED	ROADWAY RELATED	WAS AN ENGINEERING STUDY REQUESTED?
P008026+0.200	720002111101	15:00	11/06/72	MON	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008026+0.220	740002050902	17:45	9/10/74	TUE	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008026+0.300	740002880505	8:00	5/05/74	SUN	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008026+0.700	720002760603	19:05	6/16/72	FRI	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.100	720001850803	6:30	8/06/72	SUN	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008027+0.200	730001850201	16:00	2/01/73	THU	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.300	720001850804	21:55	8/15/72	TUE	U.S. NUMBERED ROUTE	INTERSECT-REL	ON ROAD	NO
P008027+0.500	720002110204	16:45	2/13/72	SUN	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.600	72R02811A001	9:30	11/08/72	WED	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.600	730001850708	14:45	7/19/73	THU	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.610	73R00508A002	21:50	8/04/73	SAT	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008027+0.620	72R00508A002	12:15	8/14/72	MON	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008027+0.650	73R00507A002	12:40	7/07/73	SAT	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.650	74R00501A003	16:00	1/23/74	WED	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO
P008027+0.700	74R00504A001	9:30	4/06/74	SAT	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008027+0.820	72R00503A001	16:20	3/02/72	THU	U.S. NUMBERED ROUTE	NON-JUNC	ON ROAD	NO
P008028+0.000	740002880802	1:00	8/19/74	MON	U.S. NUMBERED ROUTE	NON-JUNC	OFF ROAD	NO

***** ACCIDENT DETAILS *****

ACCIDENT NO.	FIRST HARMFUL EVENT	FIRST OBJECT HIT	WEATH COND	ROAD COND	LIGHT COND	TRAFFIC CONTROLS	COLLISION TYPE	INJURY SEVERITY
720002111101	MV IN TRANS	NO OBJECT HIT	CLEAR	DRY	DAYLIGHT	NO SIGNALS	HEAD ON	NON-INCAP.
740002050902	MV IN TRANS	NO OBJECT HIT	OTHER	DRY	DAYLIGHT	PAVEMENT MARKINGS	REAR END	POSS. INJ
740002880505	FIXED OBJCT	OTHER OBJECT	CLEAR	DRY	DAYLIGHT	NO SIGNALS	OTHER	INCAP. INJ
720002760603	OVERTURNED	CUT SLOPE	RAIN	WET	DAYLIGHT	NO SIGNALS	OTHER	NO INJURY
720001850803	MV IN TRANS	NO OBJECT HIT	CLEAR	DRY	DAWN/DUSK	NO SIGNALS	REAR END	NON-INCAP.
730001850201	NON-COLL.	NO OBJECT HIT	CLEAR	ICY	DAYLIGHT	NO SIGNALS	OTHER	NON-INCAP.
720001850804	MV IN TRANS	NO OBJECT HIT	CLEAR	DRY	DARK,UNLIT	NO SIGNALS	SIDE PASS	NO INJURY
720002110204	PARKED MV	OTHER OBJECT	OTHER	ICY	DAYLIGHT	NO SIGNALS	REAR END	NO INJURY
72R02811A001	FIXED OBJCT	CUT SLOPE	CLEAR	SNOWY	DAYLIGHT	NO SIGNALS	OTHER	NO INJURY
730001850708	OVERTURNED	ROCK/BOULDER	CLEAR	DRY	DAYLIGHT	NO SIGNALS	OTHER	NO INJURY
73R00508A002	ANIMAL	NO OBJECT HIT	CLEAR	DRY	DARK,UNLIT	NO SIGNALS	OTHER	NO INJURY
72R00508A002	NON-COLL.	NO OBJECT HIT	UNK	UNK	UNKNOWN	NO SIGNALS	OTHER	INCAP. INJ
73R00507A002	FIXED OBJCT	OTHER OBJECT	CLEAR	DRY	DARK,LIT	NO SIGNALS	OTHER	NO INJURY
74R00501A003	FIXED OBJCT	ROCK/BOULDER	SNOW	SNOWY	DAYLIGHT	NO SIGNALS	OTHER	NO INJURY
74R00504A001	MV IN TRANS	NO OBJECT HIT	SNOW	ICY	DAYLIGHT	NO SIGNALS	SIDE MEET	NO INJURY
72R00503A001	MV IN TRANS	NO OBJECT HIT	SNOW	ICY	DAWN/DUSK	NO SIGNALS	HEAD ON	NO INJURY
740002880802	FIXED OBJCT	TREE	CLEAR	DRY	DARK,UNLIT	NO SIGNALS	OTHER	NO INJURY

Figure 2-18. RURAL-ACC-ANALYSIS Program. Page 3 of 4.

***** ACCIDENT DETAILS *****

ACCIDENT NUMBER	TYPE	NO.	****DRIVER**** AGE	SEX	ARREST	********CONTRIBUTING-CIRCUMSTANCES******* VISION	ROAD	MECH	POSS-VIOLATION	INTENT	BODY	TRAILER	VEH YR	DAMAGE
720002111101	VEH	1	23	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	62	DISABLING
	VEH	2	47	M	YES	NOT OBSC	NO DEFECTS	NO	OTHER	GO STRAIGHT	PASS CAR	NO TRLR	73	DISABLING
	VEH	3	52	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PICKUP	NO TRLR	71	OTHER
740002050902	VEH	1	72	M	YES	NOT OBSC	NO DEFECTS	NO	IMPROPER PASS	OVERTAKE	PASS CAR	NO TRLR	69	DISABLING
	VEH	2	35	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	TRUCK	SEMI	66	NONE
740002880505	VEH	1			NO	NOT OBSC	NO DEFECTS	NO	HAD BEEN DRINK	GO STRAIGHT	PASS CAR	NO TRLR	63	DISABLING
720002760603	VEH	1	19	M	YES	NOT OBSC	SHOULDER	NO	SPEED TOO FAST	GO STRAIGHT	PASS CAR	NO TRLR	64	FUNCTIONAL
720001850803	VEH	1	22	F	NO	OTHER	NO DEFECTS	NO	OTHER	GO STRAIGHT	PASS CAR	NO TRLR	71	DISABLING
	VEH	2	22	M	NO	OTHER	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	62	DISABLING
730001850201	VEH	1	57	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	72	NONE
720001850804	VEH	1	57	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	72	OTHER
	VEH	2	19	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	LEFT TURN	PASS CAR	NO TRLR	65	OTHER
720002110204	VEH	1	22	M	NO	OTHER	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	66	DISABLING
	VEH	2			NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	REMAIN PARKED	PASS CAR	NO TRLR	66	OTHER
72R02811A001	VEH	1	63	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	68	DISABLING
730001850708	VEH	1	41	M	NO	NOT OBSC	NO DEFECTS	YES	NO APP. VIOL.	GO STRAIGHT	CNST MACH	NO TRLR		OTHER
73R00503A002	VEH	1	54	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	71	FUNCTIONAL
72R00508A002	VEH	1	73	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	MBUS/VAN	NO TRLR	66	NONE
73R00507A002	VEH	1	52	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	64	DISABLING
74R00501A003	VEH	1	62	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	73	FUNCTIONAL
74R00504A001	VEH	1	61	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	68	DISABLING
	VEH	2			NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	UNSTATED	NO TRLR		NONE
72R00503A001	VEH	1	23	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	SLOW/STOP	PASS CAR	NO TRLR	72	DISABLING
	VEH	2	49	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	REMAIN STOP	PICKUP	NO TRLR	67	DISABLING
740002880802	VEH	1	17	F	YES	NOT OBSC	NO DEFECTS	NO	SPEED TOO FAST	GO STRAIGHT	PASS CAR	NO TRLR	65	DISABLING

Figure 2-18. RURAL-ACC-ANALYSIS Program. Page 4 of 4.

The HIGH-ACC-INTERSECTNS Program

HIGH-ACC-INTERSECTNS provides a means of municipal accident analysis. It utilizes city coordinates to locate accidents. When a municipal accident occurs, the x- and y-coordinates of either the accident location or of the nearest intersection are coded in the accident data. A "grid table" is also stored in a computer file that provides the x- and y-coordinates of each intersection in the city.

When HIGH-ACC-INTERSECTNS is used, a "square-size" value is specified that indicates the size in coordinate units of each intersection. Let "s" be the square-size value, "X" and "Y" be the coordinates of an intersection, and "x" and "y" be the coordinates of an accident. The program considers the accident to have occurred at the intersection if x falls between "X - ½s" and "X + ½s" inclusive and if y falls between "Y - ½s" and "Y + ½s" inclusive. If a large enough square-size value is specified, some accidents may be assigned to two or more intersections.

Based on the square-size value, HIGH-ACC-INTERSECTNS assigns each accident to the intersection(s) in which it occurred. Some accidents will be assigned to just one intersection, some to more than one intersection, and some to no intersections. The program then counts the number of accidents that occurred at each intersection. It then prints the intersections chosen as intersections having high numbers of accidents. The method of choosing these high-accident intersections can be one of the following:

1. Print the "n" intersections having the most accidents.
2. Print all intersections having "n" or more accidents.

When the first method is used, the program is said to run in "#-intersections Mode." When the second method is used, the program is said to run in "#-accidents Mode." The program can also use a third method in which the user specifies particular intersections he wants to examine, in which case the program is said to run in "individual intersections" mode.

The program can be instructed to include all accidents, or to include only those accidents labeled as intersection accidents or intersection-related accidents (omitting non-intersection and driveway-access accidents).

To use the intersection-analysis program, prepare a command that specifies the HIGH-ACC-INTERSECTNS program. Include a CITY parameter that identifies the city be analyzed (see table 2-1 earlier in this chapter). Also include a SQUARE-SIZE parameter that specifies the square-size value in coordinate units. If desired, include the following parameter:

$$\text{ACCIDENTS=} \left\{ \begin{array}{l} \text{ALL} \\ \underline{\text{INTERSECTION}} \end{array} \right\}$$

ACCIDENTS=INTERSECTION specifies that only intersection and intersection-related accidents are included, and is the default if the ACCIDENTS parameter is omitted. Any of the following selection parameters can be included on the command (these parameters are described earlier in this chapter in the section "Commands - Accident Subsystem"):

START-DATE and END-DATE
START-ACCIDENT and END-ACCIDENT
SELECT-DD and SELECT-SIZE

If you wish to run in #-intersections mode, include the parameter

#-INTERSECTIONS=n

The "n" intersections having the highest number of accidents are printed (n may range from 1 to 999). If you wish to run in #-accidents mode, include the parameter

#-ACCIDENTS=n

All intersections having n or more accidents are printed (n may range from 1 to 999).

Figure 2-19 shows a sample HIGH-ACC-INTERSECTNS run. For this run, the #-intersections mode was used with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACCM
//SYSIN DD *
:HIGH-ACC-INTERSECTNS,CITY=BILLINGS,SQUARE-SIZE=16,
:   START-DATE=Ø7/Ø1/74,END-DATE=Ø7/31/74,
:   ACCIDENTS=ALL,#-INTERSECTNS=2
/*
```

AVE C & 13 W

ACCIDENT NUMBER	X COORD	Y COORD	DATE	TIME	DAY	NO. INJ	NO. FAT	NO. PED	NO. VEH	FIRST HARMFUL EVENT	COLLISION TYPE	I-IR	WEATH COND	ROAD COND	LIGHT CONDITION	TRAFFIC CONTROLS
740103840545	0637	0843	7/04/74	2245	THU				2	MV IN TRANS	ANGLE	I	CLEAR	DRY	DARK,LIT	TRAFFIC SIGNAL
740106160565	0685	0845	7/11/74	1718	THU				2	MV IN TRANS	ANGLE	DA	CLEAR	DRY	DAYLIGHT	NO SIGNALS

AVE C & 13 W

ACCIDENT NUMBER	TYPE	NO.	****DRIVER**** AGE	SEX	ARREST	********CONTRIBUTING-CIRCUMSTANCES******* VISION	ROAD	MECH	POSS-VIOLATION	INTENT	BODY	TRAILER	VEH YR	DAMAGE
740103840545	VEH	1	13	M	YES	NOT OBSC	NO DEFECTS	NO	OTHER	GO STRAIGHT	PASS CAR	NO TRLR	67	DISABLING
	VEH	2	17	F	NO	NOT OBSC	NO DEFECTS	YES	NO APP. VIOL.	LEFT TURN	PICKUP	NO TRLR	60	FUNCTIONAL
740106160565	VEH	1	21	M	YES	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	LEFT TURN	PASS CAR	NO TRLR	74	DISABLING
	VEH	2	23	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	74	DISABLING

----- THERE WERE 2 ACCIDENTS AT INTERSECTION AVE C & 13 W

BROADWATER & 1 W

ACCIDENT NUMBER	X COORD	Y COORD	DATE	TIME	DAY	NO. INJ	NO. FAT	NO. PED	NO. VEH	FIRST HARMFUL EVENT	COLLISION TYPE	I-IR	WEATH COND	ROAD COND	LIGHT CONDITION	TRAFFIC CONTROLS
740102660545	1016	0951	7/03/74	2115	WED				2	MV IN TRANS	REAR END	IR	CLEAR	DRY	DAWN/DUSK	NO SIGNALS
740106600594	1019	0951	7/13/74	1255	FRI	1			2	MV IN TRANS	REAR END	NJ	CLEAR	DRY	DAYLIGHT	PAVEMENT MARKINGS

BROADWATER & 1 W

ACCIDENT NUMBER	TYPE	NO.	****DRIVER**** AGE	SEX	ARREST	********CONTRIBUTING-CIRCUMSTANCES******* VISION	ROAD	MECH	POSS-VIOLATION	INTENT	BODY	TRAILER	VEH YR	DAMAGE
740102660545	VEH	1	18	F	YES	NOT OBSC	NO DEFECTS	NO	OTHER	GO STRAIGHT	PASS CAR	NO TRLR	67	FUNCTIONAL
	VEH	2	20	M	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	GO STRAIGHT	PASS CAR	NO TRLR	72	FUNCTIONAL
740106600594	VEH	1	17	M	YES	NOT OBSC	NO DEFECTS	NO	OTHER	GO STRAIGHT	MOT CYCLE	NO TRLR	73	FUNCTIONAL
	VEH	2	21	F	NO	NOT OBSC	NO DEFECTS	NO	NO APP. VIOL.	SLOW/STOP	PASS CAR	NO TRLR	68	OTHER

----- THERE WERE 2 ACCIDENTS AT INTERSECTION BROADWATER & 1 W

Figure 2-19. HIGH-ACC-INTERSECTNS Program.

To use the program in individual intersections mode, omit both the #-ACCIDENTS and #-INTERSECTIONS parameters. Include one or more data cards that specify the names of intersections. The names must correspond exactly to the names stored in the grid table. Enter the data cards via a DD statement named INTSECTN as shown in the following example job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACCM
//SYSIN DD *
:HIGH-ACC-INTERSECTNS,...
/*
//INTSECTN DD *
  One or more data cards containing intersection name in columns 1-4Ø
/*
```

The grid table is stored as a separate file in the computer. A listing of the table can be obtained by using programs described in chapter 11 of this manual.

HIGH-ACC-INTERSECTNS accesses the following files:

DD Statement	File
ACIDENT	Accident detail file - Always accessed.
ACCVEH	Accident vehicle file - Always accessed.
GRIDTBL	Grid table - Always accessed.
SCRATCH4	Scratch file - Always accessed (used for storing a temporary grid table).
TABLES	Tables - Always accessed (city and select tables).
DIRECI	Scratch file - Always accessed (used for a grid directory file that allows counting of accidents at each intersection).
DIRTIN,DIRTOUT,DIRTWKØ1,DIRTWKØ2,DIRTWKØ3	Scratch files used to sort the grid directory file.
INDXIN,INDXOUT	Scratch file for grid index that can be used to locate those intersections with the highest number of accidents.
INDXWKØ1,INDXWKØ2,INDXWKØ3	Scratch files used when sorting the grid index.
SORTLIB	Sort library - Always accessed.
SYSOUT	Printed output from sort program - Always accessed.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

The TA-1 Program

The TA-1 program is an aid in producing the annual TA-1 report. Each TA-1 command generates five sets of summaries, one each for the following systems:

1. Federal aid interstate system.
2. Federal aid primary system.
3. Federal aid secondary system.
4. Federal aid urban system.
5. Not on federal aid system.

Each of the five sets of summaries contains three summaries, one each for the following accident categories:

1. Rural accidents (accidents that occurred outside the city limits of any incorporated cities).
2. Urban accidents (cities over 5ØØØ).
3. Urban accidents (cities under 5ØØØ).

Each summary shows:

1. Number of property damage accidents.
2. Number of injury accidents.
3. Number of fatal accidents.
4. Total number of accidents (1+2+3).
5. Number of fatalities.
6. Number of injuries.

To use the program, prepare a command that specifies the TA-1 program. Include the parameter DATA=FEDERAL-AID. Also include the parameters START-DATE and END-DATE (these are described earlier in this chapter in the section "Commands - Accident Subsystem"). Any of the print parameters described in chapter 1 can be included on the command.

Figure 2-2Ø shows the last of five pages printed by a TA-1 command. The job setup for this run was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACC
//SYSIN DD *
:TA-1,DATA=FEDERAL-AID,START-DATE=Ø1/Ø1/75,END-DATE=12/31/75
/*
```

STATE OF MONTANA -- TA-1 REPORT

REPORTING PERIOD FROM 01/01/75 TO 12/31/75

NOT ON FEDERAL AID SYSTEM

RURAL ACCIDENTS

NUMBER OF PROPERTY DAMAGE ACCIDENTS	1988
NUMBER OF FATAL ACCIDENTS	36
NUMBER OF INJURY ACCIDENTS	901
TOTAL NUMBER OF ACCIDENTS	2925
NUMBER OF FATALITIES	40
NUMBER OF INJURIES	1376

URBAN ACCIDENTS (CITIES OVER 5000)

NUMBER OF PROPERTY DAMAGE ACCIDENTS	3857
NUMBER OF FATAL ACCIDENTS	9
NUMBER OF INJURY ACCIDENTS	882
TOTAL NUMBER OF ACCIDENTS	4748
NUMBER OF FATALITIES	9
NUMBER OF INJURIES	1194

URBAN ACCIDENTS (CITIES UNDER 5000)

NUMBER OF PROPERTY DAMAGE ACCIDENTS	896
NUMBER OF FATAL ACCIDENTS	5
NUMBER OF INJURY ACCIDENTS	190
TOTAL NUMBER OF ACCIDENTS	1091
NUMBER OF FATALITIES	5
NUMBER OF INJURIES	233

Figure 2-2Ø. TA-1 Program.

The TA-1 program accesses the following files:

DD Statement	File
ACCDIRI	Accident directory file - always accessed.
ACIDENT	Accident detail file - always accessed.
ROADLOG	Roadlog file - always accessed.
TABLES	Tables - always accessed.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

The Accident-by-Section Programs

The accident-by-section programs utilize a file called the accident-by-sections file. This file is constructed by programs described in the publication Highway Information System Release 4.∅ - System Maintenance. To use the accident-by-sections software you must first have system maintenance personnel generate this file for you. When you have this file built, you will be asked to specify the following options:

1. What route system(s) or route number(s) do you need (eg., Federal aid interstate system, federal aid primary route number 8).
2. What level of accidents do you wish to appear in your reports: all accidents, only those accidents that were legally reportable, or only those accidents that were not legally reportable.

Each entry in the accident-by-sections file contains the following data items:

Route system and route number
Milepoint at beginning and end of section
Remark code (identifies section type)
Description
Section length
Data for each of three years including:

Year
Adjusted vehicle miles
Adjusted section length
Number of accidents
Number of injury and fatality accidents
Number of injuries and fatalities

Earliest effective date
Latest effective date
Number of lanes
City number

Totals records are also stored that contain the total values for each route and for each route system.

Section breaks for the accident-by-sections file are taken from the traffic file. Sections are identified by type:

Rural section (type=W)
Municipal section (type=T)
Out-of-state section (type=O)
Non-existent section (type=N)
Coincident section (type=C)

Accident counts are stored only for rural sections.

The basic statistic that is provided for each rural section in the accident-by-sections report is the accident rate. The accident rate is the number of accidents per million vehicle miles. A number of complications arise in sections in which major reconstruction has occurred during the report period and in sections that were built and opened during the report period. For both of these types of situations, the roadlog file indicates the "effective" date of the roadway (the date the road was opened after construction or major reconstruction). Accidents are counted in a section only if the following conditions hold:

1. The accident occurred within the report section.
2. The accident occurred during the report period.
3. The accident date does not predate the roadlog's effective date for the section.

The roadlog file may contain several entries for a given section, so that the effective date may vary through the accident-by-sections section. Comparisons against roadlog effective dates are based on the roadlog data stored at the actual accident location.

In general, an accident that occurred at exactly the break between two report sections are assigned to the second section. Similarly, an accident that occurred at exactly the break between two roadlog sections are assigned to the second section. There are two exceptions to this rule:

1. An accident that occurred at exactly the end of a route, at exactly the beginning of a coincident section, or at exactly the beginning of a spur or loop is assigned to the last previous report section on the route.
2. An accident that occurred at exactly the city limits where a route enters a city is assigned to the last rural report section preceding the city limits.

The vehicle-miles are computed from the traffic file. Again, the calculations are complicated by roadways that were constructed or reconstructed during the report period. The ADT stored in the traffic file for this type of roadway is not prorated based upon the portion of the year during which the roadway was open. The vehicle miles is first calculated for each roadlog section within a report section without taking effective dates into account. The "effective" or "adjusted" vehicle miles is then computed by summing the separate vehicle miles under the following constraints for each roadlog section:

1. If December 31st of the report year predates the effective date of the roadlog section, the vehicle miles for the section becomes zero.
2. If January 1st of the report year postdates the effective date, the vehicle miles for the section is not adjusted.
3. If the effective date occurred during the report year, the vehicle miles is adjusted by multiplying by the ratio of the number of days in the year after the effective date and the number of days in the year.

The vehicle miles just computed is a daily vehicle miles. The annual vehicle miles is computed by multiplying by the number of days in the year. The accident rate during a report year is the number of accidents divided by the annual vehicle miles and multiplied by one million.

An adjusted section length is also utilized for computing ADT from the adjusted vehicle miles. This section length is computed in much the same manner as the adjusted vehicle miles is computed from vehicle miles.

The process just described for computing adjusted vehicle miles and adjusted section lengths to compute AADT is somewhat complicated, but is felt to be the best method. For "old" sections of highway, the procedure gives exactly the same result as the "weighted" AADT's traditionally used. For report sections involving new construction or reconstruction, this process gives meaningful results more often than any other method considered. The accident-by-sections printout includes

the effective dates used in the computations so that sections in which the adjustment was performed can be readily identified.

The LIST-ACC-BY-SECTN Program - This program prints a listing of the contents of the accident-by-sections file. Six sections are printed per page. All of the information stored in each record is printed. This program can be utilized only after the accident-by-sections file has been constructed. To use the program, prepare a command that specifies the LIST-ACC-BY-SECTN program. A DATA parameter must be included on the command (this parameter is described in chapter 1). Since the accident-by-sections file can be built for individual routes or route systems, you must know which routes are currently stored in the file. The START-MILEPOINT and END-MILEPOINT parameters can be included on the command if desired. Any of the print parameters described in chapter 1 can be included on the command.

Figure 2-21 shows a sample output from the LIST-ACC-BY-SECTN program. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACCA
//SYSIN DD *
:LIST-ACC-BY-SECTN,DATA=PRIMARY=8,END-MILEPOINT=Ø3Ø+Ø.ØØØ
/*
```

The LIST-ACC-BY-SECTN program accesses the following files:

DD Statement	File
ACCSECT	Accident-by-sections file - always accessed.
PRINTER	Printed output (this name can be changed by coding a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

```
HIS LIST-ACC-BY-SECTN

   START-KEY      REM       END-KEY                       DESCRIPTION                  LENGTH      DATE       DATE   LN  CITY
P00008000+0.000        P00008003+0.643 JCT I 90 GARRISON INT                           3.642     / /        / /    2
--------------------------------------------*-----------------------------------------------*--------------------------------------
YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT
72    4497    3.642  7   3       8      73   4257    3.642   5   2       2      74   4326    3.642   4


   START-KEY      REM       END-KEY                       DESCRIPTION                  LENGTH      DATE       DATE   LN  CITY
P00008003+0.643        P00008013+0.007 JCT FAS 407 POWELL CO                           9.375     / /        / /    2
--------------------------------------------*-----------------------------------------------*--------------------------------------
YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT
72   12600    9.375 12   4       6      73  12032    9.375  10   2       4      74  11728    9.375   8   3   1   6   1


   START-KEY      REM       END-KEY                       DESCRIPTION                  LENGTH      DATE       DATE   LN  CITY
P00008013+0.007        P00008025+0.581 JCT FAS 272 POWELL CO                          12.563     / /     12/15/73  2
--------------------------------------------*-----------------------------------------------*--------------------------------------
YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT
72   13484   10.200 12   7   1  10   1  73  14463   10.308  13   8       9      74  15761   12.563  11   3       4


   START-KEY      REM       END-KEY                       DESCRIPTION                  LENGTH      DATE       DATE   LN  CITY
P00008025+0.581        P00008027+0.167 BEG 4LNU POWELL CO                              1.516  12/15/73   12/15/73  4
--------------------------------------------*-----------------------------------------------*--------------------------------------
YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT
72                                      73    103     .069                      74   1884    1.516   2   2       5


   START-KEY      REM       END-KEY                       DESCRIPTION                  LENGTH      DATE       DATE   LN  CITY
P00008027+0.167        P00008027+0.617   POWELL COUNTY                                  .450     / /     12/15/73  4
--------------------------------------------*-----------------------------------------------*--------------------------------------
YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT
72    413     .298   2                  73    449     .304   2                  74    573     .450


   START-KEY      REM       END-KEY                       DESCRIPTION                  LENGTH      DATE       DATE   LN  CITY
P00008027+0.617        P00008039+0.654 POWELL CO LN LEWIS & CLARK CO                  11.912     / /      5/23/71  2
--------------------------------------------*-----------------------------------------------*--------------------------------------
YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT YR   AVM    ASECL   ACC IAC FAC INJ FAT
72   17594   11.912 17   8      13      73  18683   11.912  18   7       8      74  17760   11.912  28  10      14
```

Figure 2-21. LIST-ACC-BY-SECTN Program.

<u>The ACCIDENT-BY-SECTIONS Program</u> - This program prints the accident-by-sections report. Before this program is used, check to be sure that the accident-by-sections file has been constructed.

ACCIDENT-BY-SECTIONS utilizes a table of economic loss values (average dollar value for property damage, injuries, and fatalities). The table is based on figures provided by the National Safety Council, and the values vary from year to year. The first page printed by ACCIDENT-BY-SECTIONS is a listing of the economic loss table. If the table is out of date, notify system maintenance personnel. The table is described in the publication <u>Highway Information System Release 4.Ø - Record Formats and Subroutines</u>.

Beginning on the second page of output is a set of one or more summaries. These summaries show the total accident rates for each route and for each route system requested on the ACCIDENT-BY-SECTIONS command.

After the total accident summary is printed the accident-by-sections report. This report shows accident rates for each section of highway (the highway sections are those used in the traffic file for the traffic-by-sections report). Earlier in this section is included information on how the various values shown in the report are computed. Beside the accident rate printout may be printed an asterisk or a double asterisk. A single asterisk indicates a section that has an accident rate lower than a computed threshold value. A double asterisk indicates a section that has an accident rate higher than a computed threshold value. The threshold values are computed separately for each section so that oftentimes when two sections have the same accident rate one may be flagged and the other may not be flagged. The basic procedures used in computing the threshold values are described in the following references:

1. Roy Jorgensen and Associates and Westat Research Analysts, "Evaluation of Criteria for Safety Improvements on the Highway," a report prepared for the United States Bureau of Public Roads Office of Highway Safety, Washington, D.C., 1966, pages 26-28.
2. Morin, Donald A., "Application of Statistical Concepts to Accident Data," Public Roads, Volume 34, No. 7, April 1967, pp 135-137.
3. Colorado Department of Highways, "procedure for Identifying Hazardous Locations," September 14, 1973.

The upper and lower limits on a given report section are calculated using the following formulas:

$$UL = R + k\sqrt{\frac{R}{M}} + \frac{1}{2M}$$

$$LL = R - k\sqrt{\frac{R}{M}} - \frac{1}{2M}$$

UL = Upper limit for accident rate

LL = Lower limit for accident rate

R = Average accident rate (accidents per million vehicle-miles) for the given federal aid system

k = a constant, the value of which determines the probability that a section with "average" accident experience will have an actual accident rate lying between LL and UL (a value of 1.96Ø is currently in use)

M = Millions of vehicle miles traveled on the section during the given time period

To use the program, prepare a command that specifies the ACCIDENT-BY-SECTIONS program. Include a DATA parameter on the command (see chapter 1). If desired, START-MILEPOINT and/or END-MILEPOINT can also be included. Any of the print parameters described in chapter 1 can be included.

Figure 2-22 on the following pages shows a sample accident-by-sections run. The first page shows the listing of the economic loss table. The second page shows a summary for federal aid primary route 8. The third page is the accident-by-sections run for the first portion of primary route 8. The run was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISACCA
//SYSIN DD *
:ACCIDENT-BY-SECTIONS,DATA=PRIMARY=8,END-MILEPOINT=Ø3Ø+Ø.ØØØ
/*
```

ACCIDENT-BY-SECTIONS accesses the following files:

*** ACCIDENT BY SECTIONS REPORT ***

TABLE OF LOSS VALUES

	YEAR	PROPERTY	INJURY	FATALITY
	1972	$480	$3,400	$82,000
	1973	$500	$3,700	$90,000
*	1974	$500	$3,700	$90,000

* VALUES NOT FOUND IN TABLE,
THOSE OF 1973 SUBSTITUTED

Figure 2-22. ACCIDENT-BY-SECTIONS Program. Page 1 of 3.

*** ACCIDENT BY SECTIONS REPORT ***

TOTALS FOR FEDERAL AID PRIMARY   ROUTE NUMBER C0008

ACCIDENTS COUNTED: ALL

REPORT YEAR	TOTAL ACCIDENTS	TOTAL FATAL ACCIDENTS	TOTAL INJURIES	TOTAL FATALITIES	TOTAL DAILY VEH-MILES	ACC. RATE	FATAL RATE
1972	101	8	80	9	154,759	1.79	15.93
1973	120	2	71	2	172,523	1.91	3.18
1974	135	3	84	4	182,030	2.03	6.02
TOTAL	356	13	235	15	509,312	1.92	8.07

Figure 2-22. ACCIDENT-BY-SECTIONS Program. Page 2 of 3.

FEDERAL AID PRIMARY ROUTE NUMBER 00008

NO. LNS	SECTION LENGTH	EARLIEST/LATEST EFFECTIVE DATES	REP. YEAR	ADT	DAILY VEHICLE MILES	ACCIDENT BREAKDOWNS PR.D.	INJ.	FAT.	TOT.	PERSONS INJ	FAT	RATES ACC.	FAT.	ECONOMIC LOSSES PROPERTY	TOTAL
000+0.000 JCT I 90 GARRISON INT															
			72	1,234	4,497	4	3	0	7	8	0	4.2**	0.0	3,360	30,560
2	3.642		73	1,168	4,257	3	2	0	5	2	0	3.2	0.0	2,500	9,900
			74	1,187	4,326	4	0	0	4	0	0	2.5	0.0	2,000	2,000
			TOTAL	1,197	13,080	11	5	0	16	10	0	3.3	0.0	7,860	42,460
003+0.643 JCT FAS 407 POWELL CO															
			72	1,344	12,600	8	4	0	12	6	0	2.6	0.0	5,760	26,160
2	9.375		73	1,283	12,032	8	2	0	10	4	0	2.2	0.0	5,000	19,800
			74	1,250	11,728	4	3	1	8	6	1	1.8	23.3	4,000	116,200
			TOTAL	1,292	36,360	20	9	1	30	16	1	2.2	7.5	14,760	162,160
013+0.007 JCT FAS 272 POWELL CO															
			72	1,321	13,484	4	7	1	12	10	1	2.4	20.3	5,760	121,760
2	12.563	12/15/73	73	1,403	14,463	5	8	0	13	9	0	2.4	0.0	6,500	39,800
			74	1,254	15,761	8	3	0	11	4	0	1.9	0.0	5,500	20,300
			TOTAL	1,321	43,708	17	18	1	36	23	1	2.2	6.2	17,760	181,860
025+0.581 BEG 4LNU POWELL CO															
			72	*** NEW ROAD ***											
4	1.515	12/15/73 12/15/73	73	1,492	103	****	NONE	****				0.0	0.0	0	0
			74	1,242	1,884	0	2	0	2	5	0	2.9	0.0	1,000	19,500
			TOTAL	1,253	1,987	0	2	0	2	5	0	2.7	0.0	1,000	19,500
027+0.167 POWELL COUNTY															
			72	1,385	413	2	0	0	2	0	0	13.2**	0.0	960	960
4	0.450	12/15/73	73	1,476	449	2	0	0	2	0	0	12.2**	0.0	1,000	1,000
			74	1,273	573	****	NONE	****				0.0	0.0	0	0
			TOTAL	1,364	1,435	4	0	0	4	0	0	7.6**	0.0	1,960	1,960
027+0.617 POWELL CO LN LEWIS & CLARK CO															
			72	1,476	17,594	9	8	0	17	13	0	2.6	0.0	8,160	52,360
2	11.912	5/23/71	73	1,568	18,683	11	7	0	18	8	0	2.6	0.0	9,000	38,600
			74	1,490	17,760	18	10	0	28	14	0	4.3**	0.0	14,000	65,800
			TOTAL	1,512	54,037	38	25	0	63	35	0	3.1**	0.0	31,160	156,760

Figure 2-22. ACCIDENT-BY-SECTIONS Program. Page 3 of 3.

DD Statement	File
ACCSECT	Accident-by-sections file - always accessed.
TABLES	Tables - always accessed.
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter).
SYSPRINT	Used for IBM error messages.

Other Programs

The Montana Department of Highways Data Processing Bureau has programmed two programs for use in rural accident analysis. One of these is designed to analyze rural intersections. The other is designed to analyze open range areas. These programs were not fully operational at the time of this writing and sample outputs were not available. Information on the programs can be obtained from the Data Processing Bureau.

CHAPTER 3

THE BRIDGE SUBSYSTEM

Introduction

The bridge file contains one entry ("record") for each bridge in the state. Each entry contains a "key" field composed of the standard HIS key - route system, route number, and milepoint. The entries can be read sequentially in order by key, or they can be accessed directly when the key is provided.

Each record in the bridge file contains the following data elements:

Key
Remark
Coincident key
Maintenance district
Construction district
Detour length
Features intersected
Minimum vertical clearance
Total horizontal clearance
Major or minor
Latitude and longitude
Inventory route (signed route number)
Facility carried
Physical vulnerability
Custodian
Year built and year improved
Number of lanes on and under structure
Design load
Bridge median
Skew
Structure flared
Navigation control, vertical clearance, and horizontal clearance
Type of service
Structure type - main and approach
Number of spans in main unit and number of approach spans
Length of longest span
Structure length
Left and right sidewalk width
Bridge roadway width and bridge deck width
Station number
Project number
Minimum vertical clearance and underclearance
Minimum lateral underclearance - left and right
Wearing surface type and wearing surface depth
Approach guardrail
Guardrail safety

Posted speed limit
Posted load limit
Condition of deck, superstructure, culverts, etc.
Estimated remaining life
Operating rating
Structural condition
Adequacy of underclearances, waterway, etc.
Approach roadway alignment
Year of needed improvement
Type of service and type of work
Length of improvement
Proposed design loading
Proposed road width
Proposed number of lanes
Design ADT
Year of design ADT
Year of roadway improvements
Type of roadway improvements
Cost of improvements
Inspection date
Structure batch serial number
Microfilm serial number
Date of update

Because the Highway Information System is an integrated file system, other HIS files can be accessed when processing bridge data. In particular, some of the bridge programs access the roadlog, traffic, and true mileage files to obtain information for reports. The bridge file is also compatible with other files such as the accident and skid test files.

Standard User Input - Bridge Subsystem

Because the Highway Information System has been designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the bridge programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISBRID
//SYSIN DD *
   One or more commands
/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed by the computer.

The job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, you can direct the output to the terminal. A sample job setup is:

```
// JOB
// EXEC HISBRID,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *

   One or more commands

/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PRINTER-DD=CRJE instead of OUTPUT-FILE=CRJE.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Bridge Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section describes parameters that can be used on several of the bridge commands.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters provide a flexible means of selecting bridges according to the route system, route number, and milepoint. They are described in detail in chapter 1 of this manual.

The MAX-#-ENTRIES Parameter - This parameter is implemented primarily as protection to the user and as an aid in program debugging. It places a limit on the number of bridges accepted on input (only bridges that meet all of the requested selection criteria are counted). For example, if the following command is submitted:

:LIST,FILE=BRIDGE,DATA=PRIMARY,MAX-#-ENTRIES=5Ø

only the first 5Ø bridges on the primary system are printed. Any number between 1 and 99999 can be coded.

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting bridges. The SELECT-DD parameter provides the capability of choosing bridges that meet selection criteria based on the contents of nearly any data element(s) in the bridge, roadlog, and traffic files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on the use of SELECT-DD and SELECT-SIZE is included in chapter 12.

The LIST Program

The LIST program prints information from the bridge file. One line is printed for each bridge. Because all of the bridge information cannot be printed on one line, several different list formats are implemented. To obtain all of the information stored for a bridge it is necessary to print the bridge once in each format. The data elements printed for the various formats are:

Format 1

- Key
- Remark
- Coincident key
- Maintenance District
- Construction district
- Detour length
- Features intersected
- Minimum vertical clearance
- Total horizontal clearance
- Major or minor
- Latitude and longitude

Format 2

- Key
- Remark
- Coincident key
- Inventory route
- Facility carried
- Physical vulnerability
- Custodian
- Year built and year improved
- Lanes on and under structure
- Design load
- Bridge median
- Skew
- Structure flare
- Navigation control
- Navigation vertical clearance
- Navigation horizontal clearance

Format 3

- Key
- Remark
- Coincident key
- Service type
- Main structure type
- Approach structure type
- Number of main spans
- Number of approach spans
- Maximum span length
- Structure length
- Left sidewalk width
- Right sidewalk width
- Bridge roadway width
- Bridge deck width
- Station number
- Project number

Format 4

- Key
- Remark
- Coincident key
- Minimum vertical overclearance
- Minimum vertical underclearance
- Minimum lateral underclearance
- Wearing surface type
- Wearing surface depth
- Approach guardrail
- Guardrail safety
- Posted speed limit
- Posted load limit
- Deck condition
- Superstructure and substructure
- Channel and channel protection
- Culvert and retaining walls
- Remaining life
- Operating rating
- Approach alignment

Format 5

- Key
- Remark
- Coincident key
- Inventory rating
- Structural condition
- Deck geometry
- Underclearance
- Safe load capacity
- Waterway adequacy
- Approach alignment rating
- Year of needed improvement
- Type of service
- Type of work
- Improvement length
- Proposed design load
- Proposed road width
- Proposed number of lanes
- Design ADT
- Year of design ADT

Format 6

- Key
- Remark
- Coincident key
- Inspection date
- Structure batch number
- Microfilm serial number
- Date of update

The program can print a list of the data elements printed during a run. Because this list is time-consuming to output at a CRJE terminal, the field list can be skipped.

To use the program, prepare a command that specifies the LIST program. Include the parameter FILE=BRIDGE. Include a DATA parameter (see chapter 1). Also include the following parameter to identify which list format you want:

$$\text{LIST=} \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ \text{ALL} \end{array} \right\}$$

If you specify LIST=ALL, all six list formats are printed. If you wish, include the parameter

$$\text{FIELD-LIST=} \left\{ \begin{array}{c} \text{NO} \\ \underline{\text{YES}} \end{array} \right\}$$

FIELD-LIST=YES (the default used if no FIELD-LIST parameter is coded) requests a listing of the fields printed in the listing. The following parameters can be included on the command as needed (these are described earlier in this chapter in the section "Commands - Bridge Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

Any of the print parameters described in chapter 1 can be included on the command.

Figure 3-1 is a sample output from LIST. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:LIST,FILE=BRIDGE,DATA=PRIMARY=7,LIST=1,FIELD-LIST=NO
/*
```

-------KEY-------	REM	COINCIDENT-KEY	MAINT DIV	CONST DIST	DETOUR LENGTH	FEATURES-INTERSECTED	VERT CLEAR FT	VERT CLEAR IN	HOR CLEAR	MAJ MIN	LATITUDE DEG	LATITUDE MIN	LONGITUDE DEG	LONGITUDE MIN
P00007 012+0.9131			11	11	78	E FORK BITTERROOT RIVER	99	99	24.0	Y	45	50.2	113	59.[illegible]
P00007 015+0.5001			11	11	78	E FK BITTERROOT RIVER	99	99	25.0	Y	45	52.0	114	1.[illegible]
P00007 018+0.3921			11	11	78	E FK BITTERROOT RIVER	99	99	24.0	Y	45	53.6	114	[illegible]
P00007 025+0.9211			11	11	78	RYE CREEK	99	99	23.8	Y	45	57.9	114	6.1
P00007 026+0.4491			11	11	78	BITTERROOT RIVER	99	99	20.5	Y	45	58.2	114	[illegible]
P00007 029+0.8941			11	11	78	TIN CUP CREEK	99	99	25.9	Y	46	.5	114	10.1
P00007 034+0.9341			11	11	78	ROCK CREEK	99	99	23.0	Y	46	5.2	114	10.2
P00007 036+0.9011			11	11	78	LICK CREEK	99	99	26.0	Y	46	6.2	114	11.1
P00007 037+0.5551			11	11	78	LOST HORSE CREEK	99	99	23.0	Y	46	6.8	114	10.8
P00007 039+0.9331			11	11	78	CAMAS CREEK	99	99	26.0	Y	46	8.5	114	10.7
P00007 041+0.3111			11	11	78	GOLD CREEK	99	99	26.0	Y	46	10.5	114	10.9
P00007 043+0.6061			11	11	78	BITTERROOT RIVER	99	99	31.8	Y	46	11.3	114	10.0
P00007 045+0.0421			11	11	78	SKALKAHO CREEK	99	99	21.4	Y	46	12.9	114	9.2
P00007 048+0.6171			11	11	78	CORVALLIS CANAL	99	99	28.0	Y	46	10.0	114	9.4
P00007 049+0.5031			11	11	78	BITTERROOT RIVER	14	11	24.0	Y	46	16.0	114	9.7
P00007 049+0.9531			11	11	78	IRRIGATION DITCH	99	99	32.0	Y	46	17.2	114	[illegible].7
P00007 050+0.3101			11	11	78	BLODGETT CREEK	99	99	28.0	Y	46	17.3	114	9.6
P00007 052+0.7921			11	11	78	SHEAFMAN CREEK	99	99	28.1	Y	46	20.4	114	3.[illegible]
P00007 054+0.5471			11	11	78	MILL CREEK	99	99	32.0	Y	46	17.0	114	9.5
P00007 057+0.0491			11	11	78	S FK BEAR CREEK	99	99	28.2	Y	46	23.2	114	8.6
P00007 058+0.3341			11	11	78	N FK BEAR CREEK	99	99	28.2	Y	46	24.2	114	8.6
P00007 059+0.7231			11	11	78	SWEATHOUSE CREEK	99	99	28.2	Y	46	26.9	114	8.1
P00007 061+0.5031			11	11	78	BIG CREEK	99	99	28.2	Y	46	29.9	114	7.4
P00007 065+0.0551			11	11	78	MCCALLA CREEK	99	99	28.2	Y	46	30.7	114	7.1
P00007 065+1.0081			11	11	78	MCCALLA CREEK	99	99	28.2	Y	46	30.9	114	7.0
P00007 065+0.1991			11	11	78	KOOTENAI CREEK	99	99	28.4	Y	46	30.9	114	7.0
P00007 082+0.5721			11	11	78	LOLO CREEK	99	99	43.0	Y	46	45.0	114	4.8
P00007 090+0.1151			11	11	1	BITTERROOT RIVER	99	99	30.0	Y	46	49.0	114	2.0
P00007 090+0.1152	R		11	11	1	BITTERROOT RIVER	99	99	30.0	Y	46	49.0	114	2.0
P00007 094+0.7321			11	11	2	CLARK FORK R & CMSTP&P RR	99	99	29.2	Y	46	52.1	113	59.2
P00007 094+0.7322	_	P00007 096+0.9121	11	11	2	CLARK FORK R & CMSTP&P RR	99	99	29.2	Y				

NUMBER OF RECORDS LISTED: 31

Figure 3-1. Bridge LIST Program.

The Bridge LIST program accesses the following files:

DD Statement	File
BRIDGE	Bridge file
ROADLOG	Roadlog file - accessed if a select statement refers to the roadlog file
TRAFFIC	Traffic file - accessed if a select statement refers to the traffic file
TABLES	Tables - accessed if SELECT-DD is specified
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The BDG-INVENTORY-LIST Program

BDG-INVENTORY-LIST prints the annual Planning and Research Bridge listing. To use the program, prepare a command that specifies the BDG-INVENTORY-LIST program. Include a DATA parameter (see chapter 1) on the command. The following parameters can be included as needed (these are described earlier in this chapter in the section "Commands - Bridge Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
MAX-#-ENTRIES

Any of the print parameters described in chapter 1 can be included on the command.

Figure 3-2 shows a sample output from the program. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:BDG-INVENTORY-LIST,DATA=PRIMARY=7
/*
```

The inventory listing requires more than the 132 columns available on the computer's printer, so that two page formats are available. The format shown in the first page of figure 3-2 is the left-hand page and the format shown in the second page is the right-hand page. If you wish to print only the left-hand pages, include

PLANNING AND RESEARCH BUREAU BRIDGE LIST

FA RTE NUMBER	MILEPOINT	REM	COINCIDENT-KEY	SIGNED ROUTE NUMBER	PROJECT-NUMBER	PROJECT STATION	CNTY	CITY	ADT	DESIGN LOAD H	S	EST. RATED CAPA-CITY	NAME OF FEATURE CROSSED
P00007	012+0.013-1			US 93	FHEC 19 C1	70620	41		788	15	00	637	E FORK BITTERROOT RIVER
P00007	015+0.550-1			US 93	FHP 19 D1	54437	41		348	15	00	[illegible]	E FK BITTERROOT RIVER
P00007	017+0.092-1			US 93	FHP 19 E1	98066	41		359	15	00	654	[illegible] FK BITTERROOT RIVER
P00007	025+0.921-1			US 93	FAP 139 D	13492	41		905	15	00	664	RYE CREEK
P00007	026+0.449-1			US 93	FAP 139 A	16161	41		888	15	00	657	BITTERROOT RIVER
P00007	029+0.834-1			US 93	FAP 139 D	5439	41		1862	15	00	664	TIN CUP CREEK
P00007	034+0.534-1			US 93	FAP 139 B	25106	41		1920	15	00	659	ROCK CREEK
P00007	035+0.591-1			US 93	FAP 139 B	23542	41		1938	15	00	664	LICK CREEK
P00007	037+0.555-1			US 93	FAP 139 B	36980	41		1947	15	00	655	LOST HORSE CREEK
P00007	039+0.933-1			US 93	FAP 139 B	49377	41		2034	15	00	[illegible]	CAMAS CREEK
P00007	041+0.311-1			US 93	FAP 139 B	59324	41		2104	15	00	658	GOLD CREEK
P00007	043+0.655-1			US 93	F 174 B	3300	41		2218	15	12	672	BITTERROOT RIVER
P00007	045+0.942-1			US 93	NRH 174 B	2770	41		3266	15	00	664	SKALKAHO CREEK
P00007	048+0.517-1			US 93	FAP 259 B 3	10572	41		4409	15	00	699	CORVALLIS CANAL
P00007	049+0.503-1			US 93	FAP 259 B 3	15245	41		3906	15	00	667	BITTERROOT RIVER
P00007	049+0.953-1			US 93	FAP 259 B 4	17580	41		3670	15	00	668	IRRIGATION DITCH
P00007	050+0.310-1			US 93	FAP 259 B 4	19516	41		3477	15	00	664	BLODGETT CREEK
P00007	050+0.792-1			US 93	FAP 259 B 4	41864	41		3224	15	00	658	SHEAFMAN CREEK
P00007	054+0.547-1			US 93	FAP 259 B 4	22065	41		2591	15	00	658	MILL CREEK
P00007	057+0.049-1			US 93	FAP 259 B 4	55070	41		2637	15	00	658	SO FK BEAR CREEK
P00007	[illegible]+0.534-1			US 93	FAP 259 B 4	61852	41		2668	15	00	662	N FK BEAR CREEK
P00007	[illegible]+0.723-1			US 93	FAP 259 A 2	3659	41		2335	15	00	653	SWEATHOUSE CREEK
P00007	[illegible]1+0.005-1			US 93	FAP 259 A 2	13585	41		2525	15	00	664	BIG CREEK
P00007	[illegible]+0.755-1			US 93	FAP 259 A 2	31762	41		2629	15	00	664	MCCALLA CREEK
P00007	[illegible]+1.008-1			US 93	FAP 259 A 2	37182	41		2654	15	00	664	MCCALLA CREEK
P00007	[illegible]+0.1[illegible]-1			US 93	FAP 259 A 2	37802	41		2664	15	00	658	KOOTENAI CREEK
P00007	[illegible]+[illegible]73-1			US 93	F 215 12	43166	32		4865	20	44	661	LOLO CREEK
P00007	[illegible]+0.115-1			US 12	F 215 16	81071	32		5582	20	44	655	BITTERROOT RIVER
P00007	[illegible]+0.115-2	P		US 12	F 215 15	81080	32		5582	20	44	655	BITTERROOT RIVER
P00007	[illegible]+0.732-1			US 12	U-UG 405 2	2882	32	81		20	16	660	CLARK FORK R & CMSTP&P RR
P00007	09[illegible]+0.732-2	L	P00007 096+0.912-1				32	81					CLARK FORK R & CMSTP&P RR

Fibure 3-2. BDG-INVENTORY-LIST Program. Page 1 of 2.

VERTICAL OVER 10' WIDTH FT	IN	ENTIRE ROAD FT	IN	UNDER FT	IN	LATERAL UNDER RIGHT FT	LEFT FT	HORIZONTAL ROAD WIDTH FT	DECK WIDTH FT	TOTAL WIDTH FT	SKEW DEG	WEAR SURF	MED	SIDEWALKS RIGHT	LEFT	NO. LANES	APPROACH NUMBER	APPROACH TYPE	MAIN NUMBER	MAIN TYPE	MAX LENGTH FT	TOTAL STRUCT LENGTH FT	YR BLT	YR IMP
								24.0	25.0	24.0	30	4131				2			3	SG	55	140	35	
								24.0	25.0	25.0	20	4131				2			2	SG	60	130	37	
								24.0	25.0	24.0		4131				2			[illegible]	SG	60	130	37	
								23.2	24.1	23.8	20	4131				2			4	TS	19	78	36	
								20.4	20.7	20.5		4131				2			2	STT	90	182	26	
								23.2	24.2	23.8		4131				2			3	TS	19	59	36	
								22.0	23.0	23.0		4131				2			3	TAM	31	95	34	
								21.4	22.1	26.0		4131				2			4	TS	19	78	34	
								22.0	23.0	23.0		4131				2			2	TPM	115	137	34	
								21.3	22.2	26.0	45	4131				2			2	TS	19	40	34	
								21.5	22.5	26.0		4131				2			4	TS	25	130	24	
								28.0	32.3	31.8		7001				2			4	SMS	63	83	49	
								21.4	22.2	21.4		4131				2			3	TS	19	58	34	
								28.0	31.5	28.0	60	4131	2			2			1	TPM	36	36	40	
14	11							24.0	28.5	24.0		7001				2			4	STT	176	592	40	
								32.2	34.0	32.0	40	4131				2			1	TS	25	27	41	
								28.3	30.1	28.0		4131				2	2	TS	1	TS	19	50	41	
								28.1	29.3	28.1	45	4131				2	2	TS	2	TS	25	90	41	
								34.0	35.8	32.0		4131				2			1	TS	25	26	41	
								26.2	30.0	28.2		4131				2	2	TS	2	TS	25	101	41	
								28.2	30.0	28.2		4131				2			2	TS	19	39	41	
								28.2	30.0	28.2		4131				2	2	TS	1	TS	21	83	41	
								23.2	30.0	28.2		4131				2	2	TS	4	TS	19	115	41	
								28.2	30.0	28.2		4131				2			2	TS	19	39	41	
								23.2	30.0	28.2		4131				2	2	TS	1	TS	19	56	41	
								26.4	30.2	29.4		4131				2	2	TS	1	TS	25	76	41	
								43.2	45.5	43.0	20	7001				2			2	P/CB	60	123	65	
								30.0	35.3	30.0	20	7001		6.0		2			4	P/CB	67	347	68	
								30.0	35.3	30.0	20	7001		6.0		2			4	P/CB	87	347	67	
				23		11.3		29.2	69.0	29.2		7001		4.0		4	3	TBM	3	SG	150	552	58	
										29.2														

Figure 3-2. BDG-INVENTORY-LIST Program. Page 2 of 2.

the parameter CHECKPOINT=1 on the command. If you wish to print only the right-hand pages, include the parameter RESTART=2 on the command.

The printed output includes city and county number (taken from the roadlog file) and average daily traffic (taken from the traffic file). If you wish to save computer time by not accessing these file, include the following parameter on the command:

$$\text{TYPE-RUN=} \left\{ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \right\}$$

TYPE-RUN=1 bypasses the roadlog file (city number and county number are not printed). TYPE-RUN=2 bypasses the traffic file (ADT is not printed). TYPE-RUN=3 bypasses both files.

BDG-INVENTORY-LIST accesses the following files:

DD Statement	File
BRIDGE	Bridge file
ROADLOG	Roadlog file - unless bypassed
TRAFFIC	Traffic file - unless bypassed
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

Defense Bridge Reporting

A computer-generated file, the bridge report file, is used in producing the defense bridge reports. This file must be constructed before any of the programs in this section are used. Information on construction of this file is included in the publication Highway Information System Release 4.Ø - System Maintenance.

The bridge report file is organized around signed route number and defense section number rather than around federal aid routes. The defense sections are contained in a cross-reference file called the "defense section cross-reference." Information on listing this file is contained in chapter 11 of this manual. Coding and updating information is included in the publication Highway Information System Release 4.Ø - Data Coding. Be sure this file is up to date before requesting that the bridge report file be built.

The bridge report file contains the following data items:

- Signed route system
- Signed route number
- Defense section number
- Section length
- Miles from start of section
- Bridge key
- Remark
- Bypassable
- City number
- ADT
- Design load
- Capacity
- Vertical clearance
- Horizontal clearance
- Number of lanes
- Total length
- Span length
- Bridge type
- Year built
- Name of feature crossed
- County number
- Physical vulnerability
- Latitude and longitude

The "signed route system" field contains a 1 for interstate system and a 2 for all other systems, and is used only as a sort field. The "signed route number" is constructed from the inventory route field of the bridge file. A detailed description of the data elements in the report file is included in the publication <u>Highway Information System Release 4.Ø - Record Formats and Subroutines</u>.

<u>The LIST-BDGREP Program</u> - LIST-BDGREP prints the contents of the bridge report file. To use the program, prepare a command that specifies the LIST-BDGREP program. If you want to list the entire file, no parameters are needed. You can skip the first "n" records by coding SKIP=n on the command. You can limit the number of records printed by coding MAX-#-ENTRIES=n on the command. Any of the print parameters described in chapter 1 can be included on the command.

Figure 3-3 is a sample output of LIST-BDGREP. This output was obtained with the following job setup:

ROAD SECTN NO.	MILES FROM START OF SECTN	BDG DES	HIGHWAY ROUTE NUMBER	BYPASS CODE	LNGTH OF SECTN	CITY	ADT (100'S)	DESIGN LOAD	EST RAT CAP	VERT CLEAR FT IN	HOR CLEAR	NO. LANES	TOTAL LENGTH	MAX SPAN LENGTH	BRIDGE TYPE	YEAR BUILT	NAME OF FEATURE CROSSED	
00005	5.34		US 2	N	32.3		10	HS 15	669		24	2	213	81	SMS	1934	YAAK RIVER	*
00005	11.59		US 2	N	32.3		14	HS 15	659		26	2	939	264	STD	1942	KOOTENAI RIVER & BN	*
00005	14.56		US 2	N	32.3	0640	26	HS 15	650		24	2	187	104	STT	1937	CALLAHAN CREEK	
00005	15.32		US 2	N	32.3		20	HS 15	681		23	2	180	65	SMS	1947	LAKE CREEK	
00005	27.89		US 2	N	32.3		21	HS 15	695		23	2	40	40	TBM	1930	CEDAR CREEK	
00005	31.13		US 2	N	32.3		45	H 20	691		66	4	26	26	SLAB	1970	PARMENTER CREEK	
00005	32.04		US 2	N	32.3	0400	73	H 20	691		68	4	26	26	SLAB	1970	FLOWER CREEK	
00010	2.94		US 2	N	88.6		34	HS 15	662		21	2	162	64	SMS	1935	BIG CHERRY CREEK	
00010	3.88		US 2	N	88.6		12	HS 15	662		23	2	40	20	TS	1936	GETNER CREEK	
00010	12.44		US 2	N	88.6		11	HS 15	647	15 00	24	2	145	145	STT	1937	LIBBY CREEK	
00010	13.78		US 2	N	88.6		11	HS 15	639		25	2	31	15	TS	1936	SWAMP CREEK	
00010	14.38		US 2	N	88.6		11	HS 15	639		25	2	31	15	TS	1936	SWAMP CREEK	
00010	15.97		US 2	N	88.6		11	HS 15	637		27	2	47	15	TS	1936	SWAMP CREEK	
00010	24.27		US 2	N	88.6		10	HS 15	673		24	2	21	21	TS	1948	MILLER CREEK	
00010	24.83		US 2	N	88.6		10	HS 15	642	15 02	23	2	184	184	STT	1938	FISHER RIVER	
00010	35.65		US 2	N	88.6		8	H 15	658		36	2	76	25	TS	1960	MCKILLOP CREEK ROAD	
00010	38.32		US 2	N	88.6		7	HS 15	664		25	2	39	19	TS	1941	FISHER RIVER	
00010	43.13		US 2	N	88.6		10	HS 15	654		28	2	39	19	TS	1938	LANG CREEK	
00010	73.03		US 2	N	88.6		13	HS 15	653		24	2	76	25	TS	1949	ASHLEY CREEK	
00010	81.52		US 2	N	88.6		18	HS 15	685		28	2	42	41	TBM	1933	ASHLEY CREEK	
00010	82.42		US 2	N	88.6		19	HS 15	655		28	2	42	42	TBM	1933	ASHLEY CREEK	
00015	0.73	J	US 2		7.4	0375	151			14 07	29	4			TBM	1936	BN RR	
00015	1.50		US 2		7.4	0375	149	H 20	645		30	2	184	92	P/CB	1966	STILLWATER RIVER	
00015	1.50	J	US 2		7.4	0375	149	H 20	645		30	2	184	92	P/CB	1966	STILLWATER RIVER	
00015	2.67		US 2	N	7.4		35	H 20	664		43	2	93	46	P/CB	1966	SPRING CREEK	
00015	3.38		US 2	N	7.4		33	HS 15	653		22	2	898	259	STD	1936	FLATHEAD RIVER	
00020			US 2		10.1		----- NO BRIDGES IN SECTION -----											
00025	3.34		US 2	N	69.9		28	HS 15	662		26	2	592	137	SMS	1938	SOUTH FORK FLATHEAD R	*
00025	7.11		US 2	N	69.9		24	HS 15	692		22	2	23	23	SLAB	1951	MARTIN CREEK	
00025	22.72		US 2	N	69.9		8	HS 15	653		26	2	117	24	TS	1950	DEER LICK CREEK	
00025	26.33		US 2	N	69.9		8	H 20	694		28	2	301	65	SMS	1956	B N RAILWAY	
00025	29.15		US 2	N	69.9		8	H 20	637		28	2	209	75	TBM	1956	BN RY	
00025	42.01		US 2	N	69.9		7	H 20	637		30	2	744	171	SMS	1968	MIDDLE FK FLATHEAD R	*
00025	44.02		US 2	N	69.9		7	HS 15	653		19	2	160	110	STT	1930	SNOWSLIDE GULCH	*
00025	45.57	J	US 2		69.9		7			13 09	35	2			STT	1929	BN RY	
00025	47.84		US 2	N	69.9		7	H 20	644		32	2	124	41	P/CB	1966	BEAR CREEK	
00025	50.93		US 2	N	69.9		7	H 20	651		36	2	27	27	SLAB	1963	DEVIL CREEK	
00025	52.85		US 2	N	69.9		7	H 20	637		32	2	111	40	P/CB	1966	BEAR CREEK	
00025	63.84		US 2	N	69.9		10	HS 15	699		24	2	141	60	TBM	1933	MIDVALE CREEK	*
00030	0.93		US 2	N	12.5		12	HS 15	645		24	2	764	240	SA	1941	TWO MEDICINE CREEK	*
00030	11.22		US 2	N	12.5		12	HS 15	685		30	2	128	46	TBM	1940	BN RY	
00035	3.13		US 2	N	3.3		25	HS 15	699		22	2	146	40	TBM	1924	BURLINGTON NORTHERN RR	
00040	5.12		US 2	N	53.9		14	H 15	662		36	2	36	19	TS	1957	WILLOW CREEK	
00040	5.53		US 2	N	53.9		14	H 15	662		36	2	39	19	TS	1957	WILLOW CREEK OVERFLOW	
00040	30.26		US 2	N	53.9		20	HS 15	688		26	2	316	132	SG	1942	CUT BANK CREEK	*
00045	0.19	U	US 2		0.4		20			24 07	30	2			SG	1960	INT I 15	

Figure 3-3. LIST-BDGREP Program.

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:LIST-BDGREP,MAX-#-ENTRIES=48
/*
```

LIST-BDGREP accesses the following files:

DD Statement	File
BDGREP	Bridge report file
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The DEFENSE-BDG-LIST Program - DEFENSE-BDG-LIST prints the defense bridge listing. To use the program, prepare a command that specifies the DEFENSE-BDG-LIST program. The entire defense listing is always printed. No parameters are needed on the command. Any of the print parameters described in chapter 1 can be included as needed.

Figure 3-4 is one page of a DEFENSE-BDG-LIST run. The job setup for this run was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:DEFENSE-BDG-LIST
/*
```

DEFENSE-BDG-LIST accesses the following files:

DD Statement	File
BDGREP	Bridge report file
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

DEFENSE SECTION 00015 LENGTH 7.427 ROUTE US 2 SYSTEM 2

MILES FROM START	------KEY-------	RF	BYP	CIT	-ADT-	DES	CAP	VERT	HORZ	# LN	TOTAL LENGTH	MAX SPAN LENG	BRID TYPE	YR BT	NAME-OF-FEATURES-CROSSED		CN TY	VUL	LATITUD	LONGITUD
.731	P00001121+0.7211	0		67	15118			1407	29.0	4			TBM	36	BN RR		15	3	48 12.3	114 18.[illegible]
1.503	P00001122+0.3341			67	14870	4	645		30.0	2	134	92	P/CB	66	STILLWATER RIVER		15	2	48 12.6	114 17.2
1.503	P00001122+0.3342	R		67	14870	4	645		30.0	2	184	92	P/CB	66	STILLWATER RIVER		15	2	48 12.6	114 17.2
2.668	P00001123+0.5071		N		3549	4	664		43.0	2	93	45	P/CB	66	SPRING CREEK		15	2	48 13.1	114 15.8
3.878	P00001124+0.7231		N		3306	3	653		22.0	2	893	259	STD	36	FLATHEAD RIVER		15	3	48 13.3	114 14.2

DEFENSE SECTION 00020 LENGTH 10.056 ROUTE US 2 SYSTEM 2

MILES FROM START	------KEY-------	RF	BYP	CIT	-ADT-	DES	CAP	VERT	HORZ	# LN	TOTAL LENGTH	MAX SPAN LENG	BRID TYPE	YR BT	NAME-OF-FEATURES-CROSSED	CN TY	VUL	LATITUD	LONGITUD

--- NO BRIDGES IN SECTION ---

DEFENSE SECTION 00025 LENGTH 69.934 ROUTE US 2 SYSTEM 2

MILES FROM START	------KEY-------	RF	BYP	CIT	-ADT-	DES	CAP	VERT	HORZ	# LN	TOTAL LENGTH	MAX SPAN LENG	BRID TYPE	YR BT	NAME-OF-FEATURES-CROSSED		CN TY	VUL	LATITUD	LONGITUD
3.843	P00001142+0.2631		N		2787	3	662		26.0	2	592	137	SMS	38	SOUTH FORK FLATHEAD R	*	15	3	48 23.0	114 4.6
5.112	P00001144+0.3181		N		2372	3	692		22.0	2	23	23	SLAB	31	MARTIN CREEK		15	2	48 23.6	114 2.3
21.72[illegible]	P00001151+0.1451		N		799	3	658		26.4	2	117	24	TS	50	DEER LICK CREEK		15	1	48 28.6	113 50.6
26.727	P00001154+0.7141		N		775	4	699		28.0	2	361	65	SMS	56	B N RAILWAY		15	2	48 26.0	113 47.9
29.1[illegible]	P00001157+0.5251		N		752	4	637		28.0	2	209	75	TBM	56	BN RY		15	2	48 25.2	113 45.1
42.011	P00001130+0.3391		N		699	4	637		30.0	2	744	171	SMS	68	MIDDLE FK FLATHEAD R	*	15	3	48 16.4	113 36.2
44.024	P00001132+0.4981		N		700	3	653		19.9	2	160	110	STT	30	SNOWSLIDE GULCH	*	15	2	48 15.6	113 34.6
45.660	P00001134+0.1251	0			700			1309	35.5	2			STT	29	BN RR		15		48 14.4	113 34.2
47.835	P00001135+0.5011		N		710	4	644		32.0	2	124	41	P/CB	66	BEAR CREEK		15	2	48 14.2	113 31.8
50.925	P00001139+0.3181		N		726	4	651		36.7	2	27	27	SLAB	63	DEVIL CREEK		15	2	48 15.7	113 28.1
52.648	P00001191+0.3471		N		736	4	637		32.0	2	111	40	P/CB	66	BEAR CREEK		15	2	43 16.2	113 26.2
69.840	P00001209+0.1781		N		1031	3	699		24.0	2	141	60	TBM	33	MIDVALE CREEK	*	18	2	48 26.3	113 13.1

Figure 3-4. DEFENSE-BDG-LIST Program.

The SUM-BY-DESIGN-LOAD Program - This program prints a summary of defense bridges by design load. To use the program, prepare a command that specifies the SUM-BY-DESIGN-LOAD program. No parameters are needed on the command. Any of the print parameters described in chapter 1 may be included as needed.

Figure 3-5 shows a sample output from SUM-BY-DESIGN-LOAD (as well as a sample output from DEFENSE-MILEAGE). The job setup for the SUM-BY-DESIGN-LOAD run was:

```
// JOB   (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:SUM-BY-DESIGN-LOAD
/*
```

SUM-BY-DESIGN-LOAD accesses the same files shown above for DEFENSE-BDG-LIST.

The DEFENSE-MILEAGE Program - DEFENSE-MILEAGE prints a summary of mileage on the defense system. To use the program, prepare a command that specifies the DEFENSE-MILEAGE program. No parameters are needed on the command. Any of the print parameters described in chapter 1 may be included on the command as needed.

Figure 3-5 shows a sample output from DEFENSE-MILEAGE (as well as a sample output from SUM-BY-DESIGN-LOAD). The job setup for the DEFENSE-MILEAGE run was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:DEFENSE-MILEAGE
/*
```

DEFENSE-MILEAGE accesses the following files:

DD Statement	File
DEFENSE	Defense section cross-reference
TRUMILE	True mileage file
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

SUMMARY OF BRIDGES BY DESIGN LOAD

INTERSTATE

UNKNOWN	H 10	H 15	HS 15	H 20	HS 20	HS 20+MOD
	13		3	295		

OTHER PRINCIPAL ROUTES

UNKNOWN	H 10	H 15	HS 15	H 20	HS 20	HS 20+MOD
7	4	59	304	101	1	3

DEFENSE MILEAGE TOTALS

FEDERAL AID INTERSTATE	1192.169
FEDERAL AID PRIMARY	4697.258
FEDERAL AID SECONDARY	54.102
FEDERAL AID URBAN	3.120
TOTAL MILEAGE:	5946.649

Figure 3-5. SUM-BY-DESIGN-LOAD and DEFENSE-MILEAGE.

The PRE-ATTACK-BDG-TAPE Program

PRE-ATTACK-BDG-TAPE prepares a data tape of the defense bridges. The format of the tape is documented in the FHWA publication "Federal-Aid Highway Program Manual - Volume 4, Planning - Chapter 7, Emergency Planning, Section 2 - Highway Defense Bridges," dated December 9, 1974.

To use the program, prepare a command that specifies the PRE-ATTACK-BDG-TAPE program. If desired, code MAX-#-ENTRIES=n to limit the number of records written to the tape. You may include the parameter:

$$\text{LIST=}\left\{\begin{array}{l}\text{YES}\\ \underline{\text{NO}}\end{array}\right\}$$

LIST=YES requests a listing of the tape file. LIST=NO is the default if the parameter is omitted. You may also include the parameter:

$$\text{TYPE-RUN=}\left\{\begin{array}{l}\text{NOTAPE}\\ \underline{\text{TAPE}}\end{array}\right\}$$

TYPE-RUN=NOTAPE suppresses the writes to the tape file. TYPE-RUN=TAPE is the default if the parameter is omitted. Any of the print parameters described in chapter 1 can be included on the command if desired.

Unless you specify TYPE-RUN=NOTAPE, you must include a DD statement named PREATACK to define the tape file. Specify a record length of 5Ø on the DD statement.

A sample job setup for PRE-ATTACK-BDG-TAPE is:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//PREATACK DD ...
//SYSIN DD *
:PRE-ATTACK-BDG-TAPE,LIST=YES
/*
```

PRE-ATTACK-BDG-TAPE accesses the following files:

DD Statement	File
BDGREP	Bridge report file
PREATACK	Output tape (unless TYPE-RUN=NOTAPE)
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The BDG-INSPECTION-TAPE Program

This program prepares the bridge inspection tape needed by FHWA annually. To use the program, prepare a command that specifies the BDG-INSPECTION-TAPE program. Include a DATA parameter. If desired, code any of the parameters START-MILEPOINT, END-MILEPOINT, and/or MAX-#-ENTRIES. You may include the following parameter:

$$\text{LIST=} \begin{Bmatrix} \text{NO} \\ \underline{\text{YES}} \end{Bmatrix}$$

LIST=YES, the default, requests a listing of the tape that is produced. You may also include the following parameter:

$$\text{TYPE-RUN=} \begin{Bmatrix} \text{NOTAPE} \\ \underline{\text{TAPE}} \end{Bmatrix}$$

TYPE-RUN=NOTAPE suppresses the writes to the tape file. TYPE-RUN=TAPE is the default. Any of the print parameters described in chapter 1 can be included on the command.

Unless you specify TYPE-RUN=NOTAPE, you must include a DD statement named BDGTAPE to identify the output tape file. Specify a record length of 8∅.

A sample job setup for BDG-INSPECTION-TAPE is:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//BDGTAPE DD ...
//SYSIN DD *
:BDG-INSPECTION-TAPE
/*
```

BDG-INSPECTION-TAPE accesses the following files:

DD Statement	File
BRIDGE	Bridge file
ROADLOG	Roadlog file
TRAFFIC	Traffic file
TRUMILE	True mileage file
BDGFILE	Scratch file
BDGTAPE	Output tape - unless TYPE-RUN=NOTAPE
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

CHAPTER 4

THE RAILROAD SUBSYSTEM

Introduction

The railroad file contains one entry ("record") for each railroad crossing in the state. Each entry contains a "key" field composed of the standard HIS key - route system, route number, and milepoint. The entries can be read sequentially in order by key, or they can be accessed directly when a key is provided.

Each record in the railroad file contains the following data elements:

- Key
- Set A:
  - Date of field survey
  - Road width
  - Road approach 1 and road approach 2:
    - Angle at 0
    - Sight distance - left and right
    - Distance OP
    - Curvature in degrees and minutes
    - Grade of road
    - Local interference
    - Vertical sight restriction
- Set B:
  - Railroad traffic - includes number and speed by train type (passenger, freight, or switch) and by frequency (daily, weekly, or seasonal)
  - Urban/rural
  - Location (verbal description)
- Set C:
  - Date of inventory
  - Operating railroad company
  - Crossing ID number
  - Branch or line name
  - Branch or line milepost
  - Number of tracks
  - Use of tracks by other than operating railroad
  - Signalization (number of wigwags, crossbucks, etc)
  - Number of traffic lanes
  - Average number of nightly trains
  - Estimated ADT

Because the Highway Information System is an integrated file system other HIS files such as the roadlog, traffic, and true mileage files can be accessed along with the railroad files to obtain information for reports.

Standard User Input - Railroad Subsystem

Because the Highway Information System was designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the railroad programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISRRX
//SYSIN DD *
   One or more commands
/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed by the computer.

The job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, you can direct the output to the terminal. A sample job setup is:

```
// JOB
// EXEC HISRRX,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *
   One or more commands
/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PRINTER-DD=CRJE instead of OUTPUT-FILE=CRJE.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Railroad Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section describes parameters that can be used on several railroad commands.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters provide a flexible means of selecting railroad crossings according to route system, route number, and milepoint. They are described in detail in chapter 1 of this manual.

The MAX-#-ENTRIES Parameter - This parameter is implemented primarily as protection to the user and as an aid in program debugging. It places a limit on the number of railroad crossing accepted on input (only crossings that meet all of the requested selection criteria are counted).

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting railroad crossings. The SELECT-DD parameter provides the capability of choosing railroad crossings that meet selection criteria based on the contents of nearly any data element(s) in the railroad, roadlog, and traffic files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on the use of SELECT-DD and SELECT-SIZE is included in chapter 12.

The LIST Program

The LIST program prints information from the railroad file. One line is printed for each crossing. Because all of the information cannot be printed on one line, three different list formats are implemented. To obtain all of the information stored for a crossing it is necessary to print the crossing once in each format. The key field is always printed in all three formats. In any listing you can choose between the sets A, B, and C shown above on page 4-1.

To use the program, prepare a command that specifies the LIST program. Include the parameter FILE=RAILROAD. Include a DATA parameter (see chapter 1). Also include the following parameter to specify which list format you want:

$$\text{LIST=} \begin{Bmatrix} \text{A} \\ \text{B} \\ \text{C} \end{Bmatrix}$$

The following parameters can be included on the command as needed (these are described earlier in this chapter in the section "Commands - Railroad Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

Any of the print parameters described in chapter 1 can be included on the command.

Figure 3-1 is a sample output from LIST using the LIST=C format. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRRX
//SYSIN DD *
:LIST,FILE=RAILROAD,LIST=C,DATA=PRIMARY=1-38
/*
```

The railroad LIST program accesses the following files:

DD Statement	File
RAILROAD	Railroad file
ROADLOG	Roadlog file - if a select statement refers to the roadlog file.
TRAFFIC	Traffic file - if a select statement refers to the traffic file.
TABLES	Tables - if a select statement is included
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	Used for IBM error messages

The RRXREP-SORT-LIST Program

RRXREP-SORT-LIST prints a listing of the railroad "report" file. The report file must be constructed before the RRXREP-SORT-LIST program can be run. The report file is built by software using data stored in the HIS files. The construction of the report file is described in the system maintenance manual.

RRXREP-SORT-LIST can print its listing according to any of the following sort hierarchies:

(1) Route system - route number - milepoint

(2) Route system - route number - ascending hazard index - milepoint

(3) Route system - descending hazard index - route number - milepoint

(4) Operating railroad - descending hazard index - route system - route number - milepoint

LISTING BY SYSTEM/ROUTE/MILEPOST

KEY	LOCATION	COUNTY	U/R	ES #ADT	RR	TRAIN VOLUME DAILY P FR SW	TRAIN VOLUME WEEKLY P FR SW	TRAIN VOLUME SEASCNL P FR SW	HAZD INDX	PROTECTION	ADJ HAZD INDX
P00001278+0.412	US2 IN SHELBY	TOOLE	V	1263	BN	2 02 000	0 00 000	0 00 000	0202.0	FLSHG LTS	40.4
P00001497+0.360	1.6 MI W OF SACO	PHILLIPS	R	1167	BN	0 00 000	0 01 000	0 00 000	0004.9	FLSHG LTS	.9
P00001540+0.300	1.5 MI NORTHWEST GLASGOW	VALLEY	R	2774	BN	0 00 000	0 02 000	0 00 000	0037.0	FLSHG LTS	7.4
P00001657+0.120	2 MI NW OF BAINVILLE	ROOSEVELT	R	741	BN	0 00 002	0 00 000	0 00 000	0031.1	FLSHG LTS	6.2
P00003299+0.713		CASCADE	R	3342	BN	0 00 000	0 04 000	0 00 000	0117.8	XBUCKS	117.8
P00003309+0.942		CASCADE	R	985	MILW	0 00 000	0 04 000	0 00 000	0035.3	XBUCKS	35.3
P00004026+0.900		CARBON	R	1924	BN	0 00 000	0 00 000	0 25 000	0008.1	XBUCKS	8.1
P00004028+0.360	2.8 MI N OF BRIDGER	CARBON	R	1932	BN	0 00 000	0 02 000	0 00 000	0020.3	XBUCKS	20.3
P00004042+0.650		CARBON	R	2365	BN	0 00 000	0 04 000	0 00 000	0032.2	XBUCKS	32.2
P00005108+0.280		FLATHEAD	R	5174	BN	0 02 000	0 00 000	0 00 000	0662.2	XBUCKS	662.2
P00005112+0.430	AT KALISPELL	FLATHEAD	U	13877	BN	0 02 000	0 00 000	0 00 000	1665.2	XBUCKS	1665.2
P00005112+0.456	MAIN ST	FLATHEAD	U	13485	BN	0 02 000	0 00 000	0 00 000	1645.1	XBUCKS	1645.1
P00006082+0.520	PARADISE	SANDERS	R	1119	BN	0 08 002	0 00 000	0 00 000	0593.0	XBUCKS	593.0
P00007031+0.650	0.5 MI N DARBY	RAVALLI	R	2368	BN	0 00 000	0 00 004	0 00 000	0055.2	XBUCKS	55.2
P00007047+0.530	JCT 1ST-PINE ST HAMILTON	RAVALLI	U	7099	BN	0 00 000	0 01 000	0 00 000	0039.3	XBUCKS	39.3
P00009040+0.570	AT GILMAN	LEWIS AND CLARK	R	466	BN	0 00 000	0 02 000	0 00 000	0003.4	XBUCKS	3.4
P00010001+0.446		CASCADE	U	9899	BN	0 00 001	0 00 000	0 00 000	0504.8	XBUCKS	504.8
P00010001+0.596		CASCADE	U	9899	MILW	0 00 003	0 00 000	0 00 000	0890.9	FLSHG LTS	178.1
P00011048+0.300		PARK	R	1119	BN	0 00 000	0 04 000	0 00 000	0029.4	NONE	43.5
P00011077+0.170		PARK	R	618	BN	0 00 000	0 04 000	0 00 000	0023.9	XBUCKS	23.9
P00013075+0.300		MADISON	R	462	BN	0 00 000	0 00 000	0 25 000	0002.0	XBUCKS	2.0
P00013077+0.000		MADISON	R	461	BN	0 00 000	0 00 000	0 25 000	0000.9	XBUCKS	.9
P00014270+0.730	JCT FAP 2 FORSYTH	ROSEBUD	U	916	BN	1 09 006	0 00 000	0 00 000	0586.2	FLSHG LTS	117.2
P00015081+0.430		FERGUS	R	4228	MILW	0 02 000	0 00 000	0 00 000	0524.2	FLSHG LTS	104.8
P00015082+0.420	MAIN ST	FERGUS	U	5783	MILW	0 02 000	0 00 000	0 00 000	0705.5	XBUCKS	705.5
P00015326+0.700		DAWSON	R	7190	BN	0 00 002	0 00 000	0 00 000	0603.9	FLSHG LTS	120.7
P00019007+0.800	COMMERCIAL ST	DEER LODGE	U	12123	BA&P	0 12 000	0 00 000	0 00 000	9455.9	XBUCKS	9455.9
P00019036+0.500		GRANITE	R	651	BN	0 00 000	0 00 000	0 25 000	0002.7	XBUCKS	2.7
P00019053+0.300		GRANITE	R	536	BN	0 00 000	0 00 000	0 25 000	0002.2	XBUCKS	2.2
P00019055+0.020		GRANITE	R	545	BN	0 00 000	0 00 000	0 25 000	0002.4	XBUCKS	2.4
P00020032+0.650	.5 MI N OF SAVAGE	RICHLAND	R	1212	BN	0 00 002	0 00 000	0 00 000	0089.6	FLSHG LTS	17.9
P00020047+0.900	4 MI S OF SIDNEY	RICHLAND	R	1835	BN	0 00 000	0 06 000	0 00 000	0054.8	FLSHG LTS	10.9
P00022014+0.150		DANIELS	R	342	BN	0 00 002	0 00 000	0 00 000	0012.9	XBUCKS	12.9
P00028070+0.170		CARBON	R	1808	BN	0 00 000	0 00 001	0 00 000	0004.1	XBUCKS	4.1
P00029023+0.080		MADISON	R	704	BN	0 00 000	0 00 000	0 25 000	0002.9	XBUCKS	2.9
P00029037+0.180		MADISON	R	939	BN	0 00 000	0 00 000	0 25 000	0002.1	XBUCKS	2.1
P00032051+0.320		DANIELS	R	5050	BN	0 00 002	0 00 000	0 00 000	0237.9	XBUCKS	237.9
P00038006+0.380		FLATHEAD	R	5059	BN	0 02 000	0 00 000	0 00 000	0586.8	FLSHG LTS	117.3
P00043000+0.350		FERGUS	R	2750	BN	0 00 000	0 04 000	0 00 000	0095.4	XBUCKS	95.4
P00043001+0.850		FERGUS	R	2063	MILW	0 02 000	0 00 000	0 00 000	0255.8	XBUCKS	255.8
P00043006+0.100		FERGUS	R	1234	MILW	0 00 000	0 04 000	0 00 000	0039.3	XBUCKS	39.3
P00047005+0.900		DEER LODGE	R	1017	BN	2 00 004	0 00 000	0 00 000	0427.1	FLSHG LTS	85.4
P00048000+0.395		BIG HORN	R	3402	BN	0 00 000	0 00 000	0 25 000	0013.1	XBUCKS	13.1

Figure 4-1. Railroad LIST Program.

The output formats of these four hierarchies are identical - the only difference is the order in which the crossings are printed.

To use the program, prepare a command that specifies the RRXREP-SORT-LIST program (be sure that the railroad report file has been built). Include the following parameter to specify one of the above sort hierarchies:

$$\text{LIST=} \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}$$

You may specify DATA, START-MILEPOINT, and/or END-MILEPOINT if you wish - if you do not, the default is DATA=ALL. You may include any of the print parameters described in chapter 1.

Figure 4-2 is a sample output from RRXREP-SORT-LIST using the LIST=1 format. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRRX
//SYSIN DD *
:RRXREP-SORT-LIST,LIST=1,DATA=PRIMARY
/*
```

The RRXREP-SORT-LIST program accesses the following files:

DD Statement	File
RRREP	Railroad report file
SORTWKØ1 SORTWKØ2 SORTWKØ3	Scratch files for sort-merge utility program
SORTLIB	Sort-merge load module library
SYSOUT	Sort-merge printed output
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

HIS RAILROAD FILE LIST = C

CARD AND KEY	INVEN. DATE	OPER. RR	CROSSING NUMBER	BRANCH NAME	MILEPOST	TRACKS M	TRACKS O	OTHER-RR USE	# CB	# SS	# WW	# FL	# G	# LNS	NNT	EST ADT
CP00001278+0.412	040474	BN	88 620J	SHELBY LINE	99.67	1	0	0	2	0	0	2	0	2	2	0
CP00001497+0.850	020674	BN	59 735A	SACO-HOGELAND	0.13	1	0	0	2	0	0	2	0	2	0	0
CP00001540+0.300	062774	BN	88 013W	AIR FORCE BASE	1.09	1	0	0	0	0	0	2	0	2	0	0
CP00001557+0.120	100973	BN	80 169T	OPHEIM	1.29	1	0	0	0	0	0	8	0	2	0	0
CP00003299+0.713	072374	BN	059822D	AUGUSTA LINE	7.27	1	0	0	2	0	0	0	0	2	0	2818
CP00003309+0.942	072974	MILW	398823U	AGAWAN	221.00	1	0	0	2	0	0	0	0	2	0	878
CP00004026+0.800	091274	BN	099280F	CLARKS FORK	18.41	1	0	0	2	0	0	0	0	2	0	1935
CP00004028+0.850	091274	BN	99 282U	CLARK FORK BR	16.36	1	0	0	2	0	0	0	0	2	0	0
CP00004028+0.900	091274	BN	099282U	CLARKS FORK	16.36	1	0	0	2	0	0	0	0	2	0	1935
CP00004042+0.650	090474	BN	060316Y	TRAIN WRECK	12.13	1	0	0	2	0	0	0	0	2	0	832
CP00004042+0.745	090474	BN	60 316Y	ROCKY FORK BR	12.13	1	0	0	2	0	0	0	0	2	0	0
CP00005103+0.230	092673	BN	059239E	SOMERS SPUR	1231.18	1	0	0	2	0	0	0	0	2	0	4680
CP00005112+0.430	092673	BN	59 373R	SOMERS	1226.00	1	5	0	2	0	0	0	0	2	0	0
CP00005112+0.496	092573	BN	059373R	SOMERS SPUR	1226.00	1	5	0	2	0	0	0	0	4	0	0
CP00006032+0.520	081974	BN	90 972P	MAIN LINE	218.47	1	0	0	2	0	0	0	0	2	4	0
CP00007031+0.650	070374	BN	91 258R	BITTERROOT	63.09	1	0	0	2	0	0	0	0	2	0	0
CP00007047+0.530	070574	BN	91 268W	BITTERROOT	47.15	0	1	0	1	0	0	0	0	2	0	0
CP00009040+0.570	072374	BN	88 170P	AUGUSTA LINE	39.89	1	0	0	2	0	0	0	0	2	0	0
CP00010001+0.446	031574	BN	088531S	MILL LINE	0.00	0	1	0	2	0	0	0	0	2	0	4930
CP00010001+0.598	072574	MILW	398783Y	MAIN LINE	197.25	1	0	0	0	0	0	2	0	2	0	4930
CP00010001+0.579	072574	MILW	398784E	MAIN LINE	197.26	1	0	0	0	0	0	2	0	2	0	4160
CP00011043+0.800	032174	BN	104338A	PARK BRANCH	5.55	1	0	0	0	0	0	0	0	2	0	1190
CP00011049+0.340	032074	BN	104317G	PARK BRANCH	4.90	1	0	0	0	0	0	2	0	2	0	1190
CP00011077+0.170	032674	BN	104260H	SHIELDS RIVER	16.55	1	0	0	2	0	0	0	0	2	0	650
CP00013075+0.800	031574	BN	090932S	RED BLUFF	9.44	0	1	0	2	0	0	0	0	2	0	469
CP00013077+0.000	031574	BN	090922L	RED BLUFF	8.14	1	0	0	2	0	0	0	0	2	0	460
CP00014270+0.780	080574	BN	87 404M	M/L	124.62	1	1	0	0	0	0	2	0	2	8	0
CP00015031+0.430	072374	MILW	398610J	NORTHERN MONT.	60.38	1	0	0	2	0	0	4	0	2	1	4000
CP00015032+0.470	072474	MILW	398628U	NORTHERN MONT.	62.05	1	0	0	2	0	0	0	0	2	1	7580
CP00015325+0.700	100173	BN	060807X	SIDNEY BR	1.39	1	0	0	0	0	0	2	0	4	0	8181
CP00019007+0.800	052174	BA&P	842389V	MAIN LINE	25.67	1	0	0	2	0	0	0	0	2	6	2150
CP00019007+0.801	052174	BA&P	842384L	MAIN LINE	25.00	2	0	0	2	2	0	2	2	2	0	5340
CP00019038+0.500	073174	BN	091384K	PHILIPSBURG BR	25.16	1	0	0	2	0	0	0	0	2	0	450
CP00019053+0.300	073074	BN	091400S	ELEPHANT SPUR	0.06	0	1	0	1	0	0	0	0	2	0	543
CP00019055+0.020	073074	BN	091365F	PHILIPSBURG BR	8.66	1	0	0	2	0	0	0	0	2	0	543
CP00020032+0.650	100373	BN	60 859P	KNIFE RIVER SPR	0.66	0	1	0	0	0	0	2	0	2	0	0
CP00020047+0.900	100973	BN	60 637F	RICHEY	15.66	1	0	0	0	0	0	4	0	2	0	0
CP00022014+0.150	101173	BN	080287D	PHEIM BRANCH	83.72	1	0	0	2	0	0	0	0	2	0	334
CP00028070+0.170	090574	BN	099276R	ROCKY FORK	43.79	0	1	0	1	0	0	0	0	2	0	1960
CP00029023+0.080	081474	BN	090933Y	RUBY VALLEY	45.47	1	0	0	2	0	0	0	0	2	0	734
CP00029037+0.180	081374	BN	090516P	RUBY VALLEY	31.60	1	0	0	2	0	0	0	0	2	0	800
CP00032051+0.320	101573	BN	080316D	OPHEIM BRANCH	98.98	1	3	0	2	0	0	0	0	2	0	1110
CP00038006+0.380	092673	BN	059345M	SOMERS SPUR	1212.69	1	0	0	0	0	0	2	0	2	1	4920

Figure 4-2. RRXREP-SORT-LIST Program.

CHAPTER 5

ROADLOG SUBSYSTEM

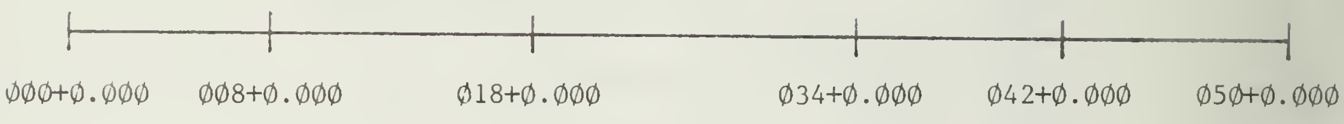
Introduction

The roadlog file contains an inventory of Montana's roadways. Each entry ("record") in the file completely describes on section of roadway, and contains the following data elements:

- Key (route system, route number, and milepoint)
- Remark
- Verbal description
- Maintenance division and maintenance section
- Project number
- County number
- Forest highway number if part of forest highway
- Location code(s̄)
- Functional classification
- Control of access
- Number of lanes
- Divided/undivided code
- City number if within city limits of an incorporated city or if in urban boundaries of an urban area
- Year built and year improved
- Surface type
- Surface and base thickness
- Surface and roadway width
- Route length
- Section length
- Constructed length
- Unimproved length
- Wye length
- Effective date
- One-way code
- Construction division
- Planning division
- Range, township, and section
- Jurisdiction
- Area name
- Map sheet

Each "mileage" record (a record that describes a section of highway) contains the above data elements. The record's key consists of the route number and route system, plus the milepoint at the beginning of the section. In addition to the mileage records a number of "descriptor" records are stored. The descriptor records identify discontinuities in routes and provide additional descriptive information as needed.

As an example of how roadlog records are stored, consider the following diagram:

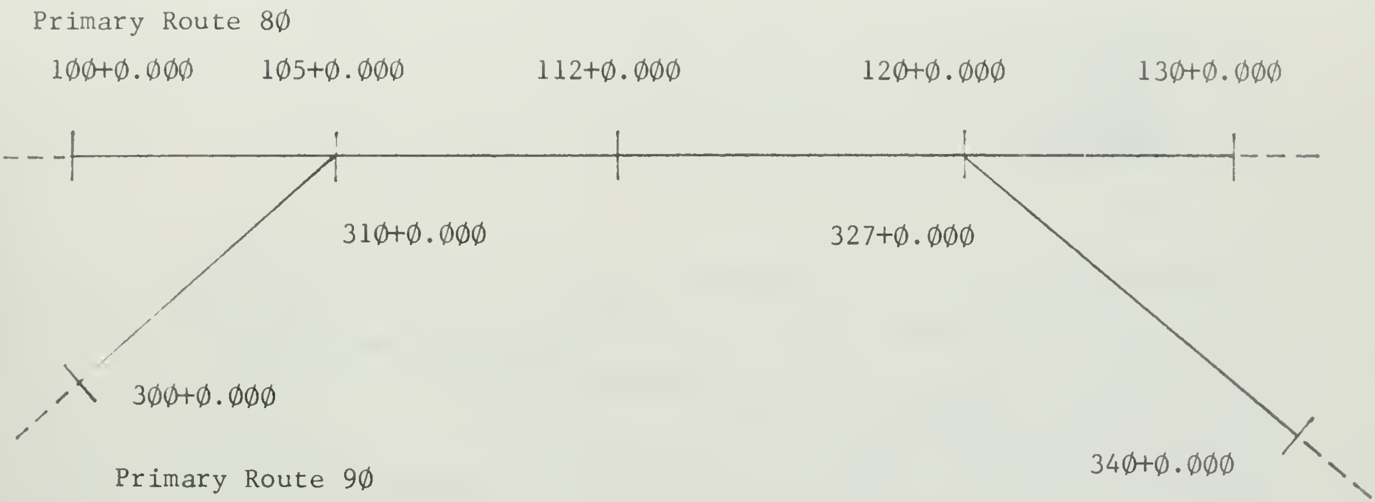


This diagram is a simple diagram of a roadway shown as a straight line. Each of the indicated milepoints is a location at which one or more roadway characteristics or jurisdictional boundaries change. For this route, the roadlog file contains a mileage record with each of the following milepoints:

ØØØ+Ø.ØØØ
ØØ8+Ø.ØØØ
Ø18+Ø.ØØØ
Ø34+Ø.ØØØ
Ø42+Ø.ØØØ

Each of these records completely describes one of the five roadway sections. As an example, the data elements stored in the record at Ø18+Ø.ØØØ are valid for the entire section with endpoints Ø18+Ø.ØØØ and Ø34+Ø.ØØØ. A sixth record, an "end-of-route" descriptor record, is stored at Ø5Ø+Ø.ØØØ. This record provides both the milepoint at the end-of-route and a verbal description of the endpoint. It does not contain any data elements to describe a section of road.

The following diagram illustrates two coincident routes:



In this diagram, primary routes 80 and 90 are shown as coincident for approximately 15 miles. In order to avoid duplication of data in the file, a cross-referencing technique is used. One of the routes is assigned the role of "base route." The base route is normally assigned to the lower-numbered route if the routes are part of the same system (in this case, route 80 is the base route) or to the higher ranking route if the routes are on different systems (for example, when an interstate route and a primary route are coincident, the interstate route is the base route). No special coding is used in the records stored for the base route. In this example, mileage records are stored at the following milepoints for route 80:

⋮

100+0.000
105+0.000
112+0.000
120+0.000
130+0.000

⋮

On the other route (the "coincident" route) is stored only those mileage records needed to describe the non-coincident sections. Route 90 contains the following records:

:

300+0.000
***
327+0.000
340+0.000

:

At 310+0.000 is stored a "coincident" descriptor record. This record indicates that route 90 is coincident with another route, and also indicates that (1) the base route is route 80, (2) the beginning milepoint on route 80 of the coincident section is 105+0.000, and (3) the ending milepoint is 120+0.000. This descriptor record is stored in the roadlog file where the '***' appears in the above list of keys. Note that only one coincident descriptor record is needed even though the base route may contain any number of records describing the coincident section.

The record type is identified by a 2-character remark code stored in the records. Each mileage record contains one of the following codes:

blank	"normal" mileage
OS	Out-of-state mileage
NE	Non-existent mileage
LP	Loop mileage
SP	Spur mileage

Each coincident record contains one of the following codes:

CO	Coincident descriptor
EN	End-of-route descriptor
ER	Additional description
DS	Additional description
IL	Interstate loop

The ER and DS descriptors simply provide additional descriptions and can be stored at any milepoints at which no other record is stored. The IL records identify the locations of interstate loops in much the same manner as coincident sections are identified.

Standard User Input - Roadlog Subsystem

Because the Highway Information System has been designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the roadlog programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISRLG
//SYSIN DD *

   One or more commands

/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed to run any programs.

This job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, you can direct the output to your terminal. A sample job setup is:

```
// JOB
// EXEC HISRLG,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *

   One or more commands

/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PARM='PRINTER-DD=CRJE' instead of PARM='OUTPUT-FILE=CRJE'.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Roadlog Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section describes parameters that are used on a number of roadlog commands.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters provide a flexible means of selecting individual routes or route systems for processing in a run. They are described in detail in chapter 1 of this manual.

The MAX-#-ENTRIES Parameter - This parameter is implemented primarily as protection to the user and as an aid in program debugging. It places a limit on the number of roadlog records accepted on input (only records that meet all of the specified selection criteria are counted).

The RLG-TYPES Parameter - This parameter provides a means of selecting records by type. Each record type is assigned a unique number:

1	Mileage records (remarks ' ','SP','LP','NE','OS')
2	Coincident descriptors (remark 'CO')
4	End-of-route descriptors (remark 'EN')
8	Interstate loop descriptors (remark 'IL')
16	Other descriptors (remark 'DS' and 'ER')
32	Any other records

Specify the sum of the appropriate values. For example, to process mileage records, coincident descriptors, and end-of-route descriptors specify RLG-TYPES=7.

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting roadlog sections. The SELECT-DD parameter provides the capability of selecting roadlog records that meet selection criteria based on the contents of nearly any data element(s) in the roadlog and traffic files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on the use of SELECT-DD and SELECT-SIZE is included in chapter 12.

The LIST Program

The LIST program prints selected fields from roadlog records. One line is printed for each roadlog section. The following data elements are printed:

Route system and route number
Milepoint (at beginning of section)
Verbal description
Project classification
Year built
Surface type
Surface width and roadway width (feet)
Location code(s)
Route length
Section length
Constructed length
Unimproved length
Wye length
Accumulated mileage (calculated by summing section length)
Remark
County number
City number

A listing of project classification codes can be found on page 5-13 of the data coding manual. Surface type codes are:

PRM	Primitive	BST	Bituminous surface treatment
BLD	Unimproved (bladed)	RMS	Road mix
GRD	Graded and drained	PMS	Plant mix
GRV	Gravel	PCC	Portland cement concrete

The location codes are:

CITY	Within city limits of incorporated city
URBN	Outside city limits but inside urban limits of urban area
CNTY	County
IRES	Indian reservation
NFOR	National forest
NMON	National monument
NPRK	National park
NREF	National wildlife refuge
MRES	Military reservation
SFOR	State forest
SPRK	State park
SGAM	State game preserve
BLM	Bureau of Land Management

A list of county numbers is given in Table 5-1 on the following page. A list of city numbers can be found in Table 2-1 of chapter 2.

To use the program, prepare a command that specifies the LIST program. Include the parameter FILE=ROADLOG. Include a DATA parameter (see chapter 1). Any of the following parameters can be coded on the command as needed (these are described earlier in this chapter in the section "Commands - Roadlog Subsystem"):

TABLE 5-1

COUNTY NUMBERS AND NAMES

County Number	County Name	County Number	County Name	County Number	County Name
Ø1	BEAVERHEAD	2Ø	GRANITE	38	POWDER-RIVER
Ø2	BIG-HORN	21	HILL	39	POWELL
Ø3	BLAINE	22	JEFFERSON	4Ø	PRAIRIE
Ø4	BROADWATER	23	JUDITH-BASIN	41	RAVALLI
Ø5	CARBON	24	LAKE	42	RICHLAND
Ø6	CARTER	25	LEWIS-AND-CLARK	43	ROOSEVELT
Ø7	CASCADE	26	LIBERTY	44	ROSEBUD
Ø8	CHOUTEAU	27	LINCOLN	45	SANDERS
Ø9	CUSTER	28	MCCONE	46	SHERIDAN
1Ø	DANIELS	29	MADISON	47	SILVER-BOW
11	DAWSON	3Ø	MEAGHER	48	STILLWATER
12	DEER-LODGE	31	MINERAL	49	SWEET-GRASS
13	FALLON	32	MISSOULA	5Ø	TETON
14	FERGUS	33	MUSSELSHELL	51	TOOLE
15	FLATHEAD	34	PARK	52	TREASURE
16	GALLATIN	35	PETROLEUM	53	VALLEY
17	GARFIELD	36	PHILLIPS	54	WHEATLAND
18	GLACIER	37	PONDERA	55	WIBAUX
19	GOLDEN-VALLEY			56	YELLOWSTONE

Note: The county numbers listed in this table are based on the alphabetical numbering system. The alphabetical numbering system is widely used in HIS, but the vehicle registration system is used in the accident subsystem.

START-MILEPOINT and/or END-MILEPOINT
RLG-TYPES
SELECT-DD and/or SELECT-SIZE
MAX-#-ENTRIES

Any of the print parameters described in chapter 1 can be included on the command.

Figure 5-1 is a sample output from LIST. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:LIST,FILE=ROADLOG,DATA=PRIMARY=17
/*
```

The roadlog LIST program accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
TABLES	Tables
TRAFFIC	Traffic file - accessed if a select statement refers to the traffic file
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The DUMP Program

DUMP prints roadlog records in the data card format used for data entry. The data card formats are shown in the data coding manual. Use of the DUMP program is identical to LIST - simply specify DUMP instead of LIST.

Figure 5-2 is a sample output of DUMP. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:DUMP,FILE=ROADLOG,DATA=PRIMARY=17,END-MILEPOINT=Ø3Ø+Ø.ØØØ
/*
```

The DUMP program accesses the same files under the same conditions as shown above for LIST.

RT-#	MILE POST	******* SECTION DESCRIPTION ********	PROJ	YR BT	SUR TYP	WIDTH SF	RD	LOCATIONS	************ LENGTH ************ ROUTE	SECTN	CONST	UNIMP	WYE	ACCUM MILEAGE	PE DS	CY	CITY
P00017		JS 10													DS		
[illegible]	[illegible]	JCT FAP 2 E OF MILES CITY	FAP	48	RMS	28	30	CNTY	6.191	6.188	6.168			6.168		9	
[illegible]	106+0.181	CUSTER COUNTY	FI	54	PMS	24	32	CNTY	2.171	2.171	2.171			8.359		9	
[illegible]	[illegible]	CUSTER COUNTY	I	62	PMS	32	32	CNTY	.659	.655	.655			9.014		9	
[illegible]	[illegible]	JCT I 94												9.014	FR		
[illegible]	[illegible]	COIN I0009+ 149+0.238-154+0.006												13.952	CO		
[illegible]	[illegible]	COIN I0009+ 154+0.006-159+0.648												19.055	CO		
[illegible]	[illegible]	JCT I 94 DIAMOND RING INT	I	71	PMS	26	26	CNTY	.579	.579	.579			19.634		9	
[illegible]	[illegible]	CUSTER COUNTY	FAP	29	BST	24	26	CNTY	3.286	3.286	3.286			22.920		9	
[illegible]	[illegible]	CUSTER CO LN PRAIRIE CO	FAP	31	RMS	22	26	CNTY	7.332	7.332				30.252		40	
[illegible]	[illegible]	PRAIRIE COUNTY	F	66	RMS	28	28	CNTY	1.561	1.561	1.561			31.813		40	
[illegible]	[illegible]	PRAIRIE COUNTY	FAP	31	RMS	24	30	CNTY	4.030	4.030				35.843		40	
[illegible]	[illegible]	AT TERRY CL SPRING ST	FAP	31	RMS	24	30	CITY	.584	.584	11.946			36.427		40	109
[illegible]	[illegible]	SPRING STREET	FAP	31	RMS	24	30	CITY	.418	.418				36.845		40	109
[illegible]	[illegible]	E/ TERRY CL	FAP	31	RMS	24	30	CNTY	.006	.006				36.851		40	
[illegible]	[illegible]	JCT FAS 253 PRAIRIE COUNTY	FAP	31	RMS	24	30	CNTY	1.288	1.288	1.712			38.139		40	
[illegible]	[illegible]	PRAIRIE COUNTY	FI	55	RMS	24	32	CNTY	.844	.844	.344			38.983		40	
[illegible]	[illegible]	PRAIRIE COUNTY	FAP	31	RMS	24	30	CNTY	.587	.587				39.570		40	
[illegible]	[illegible]	PRAIRIE COUNTY	FAP	34	RMS	22	32	CNTY	.303	.303	.303			39.873		40	
[illegible]	[illegible]	PRAIRIE COUNTY	FAP	31	RMS	24	24	CNTY	4.230	4.230	4.817			44.103		40	
[illegible]	[illegible]	PRAIRIE COUNTY	FI	49	RMS	24	32	CNTY	.878	.878	.878			44.981		40	
[illegible]	[illegible]	PRAIRIE COUNTY	F,I	49	RMS	24	32	CNTY	.163	.163	.163			45.144		40	
P00017	[illegible]	JCT I 94 AT FALLON												45.144	FR		
[illegible]	[illegible]	COIN I0009+ 184+0.893-191+0.081												51.471	CO		
[illegible]	[illegible]	COIN I0009+ 191+0.081-203+0.878												64.085	CO		
[illegible]	[illegible]	COIN I0009+ 203+0.878-209+0.870												70.265	CO		
[illegible]	[illegible]	COIN I0009+ 209+0.870-210+0.371												70.637	CO		
[illegible]	[illegible]	LOOP I0009+ 141+0.153-149+0.238												70.637	IL		
[illegible]	[illegible]	LOOP I0009+ 159+0.648-184+0.893												70.637	IL		
[illegible]	[illegible]	JCT I 94 W GLENDIVE INT												70.637	EN		

NUMBER OF RECORDS LISTED: 30

Figure 5-1. Roadlog LIST Program.

```
HIS ROADLOG FILE DUMP

AP00017           US 10                                                    DS

AP00017000+0.000JCT FAP 2 E OF MILES CITY           910FAP 158 A      9  36 3  4
BP00017000+0.0002     48484231202002830 6191 6183          6188 51671      52

AP00017006+0.191  CUSTER COUNTY                     910FI 158 8       9   2 3  4
BP00017006+0.1912     54544251201402432 2171 2171          2171            52

AP00017008+0.358  CUSTER COUNTY                     910I 94 4 6       9  24 3  4
BP00017008+0.3582     62624251301403232  659  655           655            52

AP00017009+0.022JCT I 94                                          ER
BP00017009+0.022                                              73071

AP00017009+0.023COIN I00094 149+0.238-154+0.006                   CO
BP00017009+0.023                                             123171

AP00017014+0.006COIN I00094 154+0.006-159+0.648                   CO

AP00017013+0.648JCT I 94 DIAMOND RING INT           910I 94 4 25      9  24 3  6
BP00017013+0.6482     71714251241002626  579  579           579123171      52

AP00017019+0.177  CUSTER COUNTY                     910FAP 158 B      9  36 3  6
BP00017019+0.1772     29413210 71202426 3286 3286          3286123171      52

AP00017022+0.374CUSTER CO LN PRAIRIE CO            4010FAP 158 C     40  36 3  6
BP00017022+0.3742     3157423130 702226 7332               7332            41

AP00017029+0.626  PRAIRIE COUNTY                   4010F 158 9       40   1 3  6
BP00017029+0.6262     66664231241702828 1561 1561          1561            41

                NUMBER OF RECORDS LISTED:    11
```

Figure 5-2. Roadlog DUMP Program.

The LIST-ILOOPS Program

The LIST-ILOOPS program prints a listing of all the interstate loops. About 15 to 2Ø pages of output are generated. Figure 5-3 shows a portion of a LIST-ILOOPS run. The run was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:LIST-ILOOPS,FILE=ROADLOG
/*
```

You may specify any of the print parameters shown in chapter 1 on the LIST-ILOOPS command. LIST-ILOOPS accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
TABLES	Tables
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The LIST-&-SUM Program

LIST-&-SUM prints a listing of selected roadlog fields plus a summary by county for each route processed. The data elements printed in the listing are:

Verbal description
Route number and system
Milepoint (at beginning of section)
Maintenance section
Project number
Year built and year improved
Surface type (these codes are shown above under LIST)
Surface and base thickness (inches)
Surface and roadway width (feet)
Number of lanes
Section length
Location code(s)
Project (constructed) length
Accumulated mileage (calculated by summing section lengths)

```
                                                                       ROADLOG INTERSTATE LOOPS

    MILE                                                 PROJECT          CY               CITY  YEARS  SURF  WIDTH  *** LENGTH ***
    POST          SECTION DESCRIPTION                    NUMBER       RE  NO   LOCATIONS  NMBR  PT IM  TYPE  SF RD  ROUTE  SECTION

INTERSTATE LOOP  #  1

   PRIMARY ROUTE 00002
 LOOP ON INTERSTATE ROUTE 00090  FROM MILEPOST 075+0.163 TO 082+0.623

     075+0.163  ALBERTON INT JCT FAP 2 ENT CL           I 90 1 17         31      1         1  63 63  4251  76 76   .434    .434  OUT
     075+0.597  END 4LND NO NAME ST                     I 90 1 17         31      1         1  63 63  4251  44 44   .950    .950  OUT
     076+0.547  LV ALBERTON CL                          I 90 1 17         31      3            63 63  4251  44 44   .824    .824  OUT
     077+0.371  MINERAL CO LN MISSOULA CO               I 90 2 13         32      3            63 63  4251  44 44  2.155   2.155  OUT
     079+0.525     MISSOULA COUNTY                      I 90 2 21         32      3            64 64  4251  44 44  2.087   2.087  OUT
     081+0.613  BEG 4LND MISSOULA CO                    I 90 2 21         32      3            64 64  4251  76 76  1.009   1.009  OUT

        *** MILEAGES FOR LOOP ***

  OUTSIDE FEDERAL RESERV.       7.459
                    URBAN       0.000
                ALL OTHER       0.000

                    TOTAL       7.459


INTERSTATE LOOP  #  2

   PRIMARY ROUTE 00002
 LOOP ON INTERSTATE ROUTE 00090  FROM MILEPOST 133+0.065 TO 154+0.280

     133+0.065  W BEARMOUTH INT JCT FAP 2               I 90 3 14         20      3            70 70  6203  76 76  4.286   4.286  OUT
     142+0.373     GRANITE COUNTY                       I 90 3 13         20      3            70 70  6203  76 76  7.589   7.589  OUT
     149+0.950     GRANITE COUNTY                       I 90 3 12         20      3            66 66  6201  76 76  3.368   3.368  OUT
     153+0.319  ENT DRUMMOND CL NO NAME ST              I 90 3 12         20      1        35  66 66  6201  76 76   .037    .037  OUT
     153+0.356  W DRUMMOND INT JCT FAP 2                I 90 3 12         20      1        35  66 66  6201  76 76   .904    .904  OUT
     154+0.260  LV DRUMMOND CL                          I 90 3 12         20      3            66 66  6201  76 76   .020    .020  OUT

        *** MILEAGES FOR LOOP ***

  OUTSIDE FEDERAL RESERV.      16.204
                    URBAN       0.000
                ALL OTHER       0.000

                    TOTAL      16.204
```

Figure 5-3. LIST-ILOOPS Program.

To use the program, prepare a command that specifies the LIST-&-SUM program. Include the parameter REPORT=ROADLOG and include a DATA parameter (see chapter 1). You may also include any of the following parameters as needed (these are described earlier in this chapter in the section "Commands - Roadlog Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
RLG-TYPES
SELECT-DD and/or SELECT-SIZE
MAX-#-ENTRIES

Figure 5-4 shows a sample output from the LIST-&-SUM program. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:LIST-&-SUM,REPORT=ROADLOG,DATA=PRIMARY=17
/*
```

LIST-&-SUM accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
TABLES	Tables
TRAFFIC	Traffic file - accessed if referred to in a select statement
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The SURF-TYPE Program

The SURF-TYPE program is a flexible program for generating summaries by surface type. It can summarize based on any of these seven data elements:

Route number
Year built
Year improved
Surface width
Project classification
City number (municipal and urban mileage only)
County number

FEDERAL AID PRIMARY ROUTE NUMBER 17

SECTION DESCRIPTION	MILE POST	MTCE SECT	PROJECT NUMBER	YEAR BLT	YEAR IMP	SURF TYPE	THICKNESS SURF	THICKNESS BASE	WIDTH SURF	WIDTH RDY	NO LN	SECT LENGTH	LOCATION	PROJCT LENGTH	ACCUM MILEAGE
				US 10											
JCT FAP 2 E OF MILES CITY	0+0.000	0910	FAP 158 A	48	48	PMS	2.0	20.0	28	30	2	6.188	COUNTY	6.188	6.188
CUSTER COUNTY	6+0.191	0910	FI 158 8	54	54	PMS	2.0	14.0	24	32	2	2.171	COUNTY	2.171	8.359
CUSTER COUNTY	8+0.358	0910	I 94 4 6	62	62	PMS	3.0	14.0	32	32	2	.655	COUNTY	.655	9.014
JCT I 94															
COIN I00094 149+0.333-154+0.006	9+0.023														13.952
COIN I00094 154+0.006-159+0.648	14+0.006														19.055
JCT I 94 DIAMOND RING INT	18+0.648	0910	I 94 4 25	71	71	PMS	2.4	10.0	26	26	2	.579	COUNTY	.579	19.634
CUSTER COUNTY	19+0.177	0910	FAP 158 B	29	41	PST	.7	12.0	24	26	2	3.286	COUNTY	3.286	22.920
CUSTER CO LN PRAIRIE CO	22+0.374	4010	FAP 158 C	31	57	RMS	3.0	7.0	22	26	2	7.332	COUNTY		30.252
PRAIRIE COUNTY	29+0.626	4010	F 158 9	65	66	RMS	2.4	17.0	28	28	2	1.561	COUNTY	1.561	31.813
PRAIRIE COUNTY	31+0.365	4010	FAP 158 C	31	57	RMS	3.0	7.0	24	30	2	4.030	COUNTY		35.843
ENT TERRY CL SPRING ST	35+0.431	4010	FAP 158 C	31	57	RMS	3.0	7.0	24	30	2	.584	CITY	11.946	36.427
SPRING STREET	36+0.136	4010	FAP 130 C	31	57	RMS	3.0	7.0	24	30	2	.418	CITY		36.845
LV TERRY CL	36+0.554	4010	FAP 130C	31	57	PMS	3.0	7.0	24	30	2	.006	COUNTY		36.851
JCT FAS 253 PRAIRIE COUNTY	36+0.560	4010	FAP 130 C	31	57	RMS	3.0	7.0	24	30	2	1.288	COUNTY	1.712	38.139
PRAIRIE COUNTY	37+0.851	4010	FI 130 16	55	55	RMS	2.0	13.0	24	32	2	.844	COUNTY	.844	38.983
PRAIRIE COUNTY	38+0.695	4010	FAP 130 C	31	57	RMS	3.0	7.0	24	30	2	.567	COUNTY		39.570
PRAIRIE COUNTY	39+0.282	4010	FAP 130 D	34	41	RMS	2.0	10.0	22	32	2	.303	COUNTY	.303	39.873
PRAIRIE COUNTY	39+0.585	4010	FAP 130 C	31	49	RMS	1.5	10.0	24	24	2	4.230	COUNTY	4.617	44.103
PRAIRIE COUNTY	43+0.815	4010	FI 130 10	49	49	RMS	2.0	8.0	24	32	2	.878	COUNTY	.878	44.981
PRAIRIE COUNTY	44+0.693	4010	FGI 130 11	49	49	RMS	2.0	8.0	24	32	2	.163	COUNTY	.163	45.144
JCT I 94 AT FALLON															
COIN I00094 184+0.333-191+0.031	44+0.856														51.471
COIN I00094 191+0.031-203+0.873	51+0.081														64.085
COIN I00094 203+0.878-209+0.370	63+0.878														70.265
COIN I00094 209+0.370-210+0.371	69+0.870														70.637
JCT I 94 W GLENDIVE INT	70+0.435														

********** SUMMARY BY COUNTIES **********

	ROUTE	CONST	UNIMP	WYE	CITY <5000	URBAN INSIDE CITY LIMITS	URBAN OUTSIDE CITY LIMITS	TOTAL CITY	TOTAL URBAN	COUNTY	NATL FOREST	INDIAN RESERV	OTHER	NET RURAL
CUSTER	12.886	12.879								12.879				
PRAIRIE	22.224	22.224			1.002			1.002		21.222				
TOTAL	35.110	35.103			1.002			1.002		34.101				34.101

Figure 5-4. LIST-&-SUM Program.

To use the program, prepare a command that specifies the SURF-TYPE program. Include the parameter REPORT=ROADLOG and include a DATA parameter (see chapter 1). Also include the following parameter:

$$\text{SUMMARY=} \left\{ \begin{array}{l} \text{RTE-NO} \\ \text{YR-BLT} \\ \text{YR-IMP} \\ \text{SUR-WD} \\ \text{PROJ-\#} \\ \text{CITIES} \\ \text{COUNTY} \end{array} \right\}$$

The summary parameter identifies which of the seven data elements are to be summarized. You may include the following parameter as needed:

$$\text{MILEAGE=} \left\{ \begin{array}{l} \text{URBAN} \\ \underline{\text{ALL}} \end{array} \right\}$$

MILEAGE=URBAN specifies that only mileage with the city limits or urban limits of an urban area is to be included. MILEAGE=ALL is the default if MILEAGE is not specified. You may specify any of the following parameters (these are described earlier in this chapter in the section "Commands - Roadlog Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
RLG-TYPES
SELECT-DD and/or SELECT-SIZE
MAX-#-ENTRIES

A special form of the DATA parameter, DATA=ILOOP, is available for the SURF-TYPE program only. It requests that interstate loop mileage be summarized. You may include any of the print parameters described in chapter 1 on the SURF-TYPE command.

Figure 5-5 shows two SURF-TYPE outputs, one by county and one by surface width The outputs were obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:SURF-TYPE,REPORT=ROADLOG,DATA=PRIMARY=17,SUMMARY=COUNTY
:SURF-TYPE,REPORT=ROADLOG,DATA=PRIMARY=17,SUMMARY=SUR-WD
/*
```

SURF-TYPE accesses the same files under the same conditions as shown above for LIST-&-SUM.

SUMMARY OF SURFACE TYPES
NET CONSTRUCTED LENGTH -- BY COUNTY

COUNTY		PRIMITIVE	UNIM-PROVED	GRADED & DRAINED	GRAVEL	BIT SURF TREAT	ROAD MIX	PLANT MIX	P.C. CONCRETE	TOTAL
CUSTER	RURAL					3.286	6.188	3.405		12.879
PRAIRIE	RURAL						21.222			21.222
	MUNIC						1.002			1.002
	TOTAL						22.224			22.224
TOTAL	RURAL					3.286	27.410	3.405		34.101
	MUNIC						1.002			1.002
	TOTAL					3.286	28.412	3.405		35.103

SUMMARY OF SURFACE TYPES
NET CONSTRUCTED LENGTH -- BY SURFACE WIDTH

SURFACE WIDTH		PRIMITIVE	UNIM-PROVED	GRADED & DRAINED	GRAVEL	BIT SURF TREAT	ROAD MIX	PLANT MIX	P.C. CONCRETE	TOTAL
22	RURAL						7.635			7.635
24	RURAL					3.286	12.026	2.171		17.483
	MUNIC						1.002			1.002
	TOTAL					3.286	13.028	2.171		18.485
26	RURAL							.579		.579
28	RURAL						7.749			7.749
32	RURAL							.655		.655
TOTAL	RURAL					3.286	27.410	3.405		34.101
	MUNIC						1.002			1.002
	TOTAL					3.286	28.412	3.405		35.103

Figure 5-5. SURF-TYPE Program.

The SUMMARY-BY-ROUTES Program

The SUMMARY-BY-ROUTES program prints a summary by route number and mileage type. One line is printed per route. To use the program, prepare a command that specifies the SUMMARY-BY-ROUTES program. Include a DATA parameter using one of these options:

```
          ⎧ INTERSTATE           ⎫
          ⎪ PRIMARY              ⎪
DATA=     ⎨ SECONDARY            ⎬
          ⎪ URBAN                ⎪
          ⎪ INTERSTATE+PRIMARY   ⎪
          ⎩ FEDERAL-AID          ⎭
```

You may also include any of the print parameters shown in chapter 1.

Figure 5-6 shows a sample output from SUMMARY-BY-ROUTES (as well as one from SUMMARY-BY-LOCATION). This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:SUMMARY-BY-ROUTES,REPORT=ROADLOG,DATA=INTERSTATE
/*
```

SUMMARY-BY-ROUTES accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The SUMMARY-BY-LOCATION Program

SUMMARY-BY-LOCATION prints a summary by route number and mileage type. To use the program, prepare a command that specifies the SUMMARY-BY-LOCATION program. Include the parameter REPORT=ROADLOG and a DATA parameter in the following format:

```
       ⎧ INTERSTATE         ⎫
DATA=  ⎨ INTERSTATE+PRIMARY ⎬
       ⎩ SECONDARY          ⎭
```

You may also include any of the print parameters described in chapter 1.

DETAIL SUMMARY BY ROUTES -- FEDERAL AID INTERSTATE SYSTEM

ROUTE NO.	NET ROUTE MILEAGE	** DEVELOPMENT STATUS ** CONSTR	UNIMP	WYE	CITY	COUNTY	NATIONAL FOREST	INDIAN RESERVE	GAME REFUGE	STATE FOREST	NATL PARKS	STATE PARKS	NATL MON	MILIT RESERVE
					******	******	******	******	LOCATED IN	******	******	******	******	******
15	400.815	400.815			9.620	379.369	11.826							
90	543.577	543.577			16.254	441.420	30.600	55.711						
94	248.787	248.787			2.941	237.302			8.544					
TOTAL	1193.179	1193.179			28.815	1058.091	42.426	55.711	8.544					

SUMMARY OF ROUTE LENGTHS AND LOCATIONS
FEDERAL AID INTERSTATE SYSTEM

ROUTE NO.	OUTSIDE FEDERAL RESERV. MUNIC	COUNTY	TOTAL	*** INSIDE FEDERAL RESERVATIONS *** NATIONAL FOREST	INDIAN RESERV.	GAME REFUGE	TOTAL	GRAND TOTAL
15	9.620	379.369	388.989	11.826			11.826	400.815
90	15.846	441.420	457.266	30.600	55.711		86.311	543.577
94	2.941	237.302	240.243			8.544	8.544	248.787
TOTAL	28.407	1058.091	1086.498	42.426	55.711	8.544	106.681	1193.179

Figure 5-6. Roadlog SUMMARY-BY-ROUTES and SUMMARY-BY-LOCATION.

Figure 5-6 shows a sample output from SUMMARY-BY-LOCATION (as well as one from SUMMARY-BY-ROUTES). This output was obtained from the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:SUMMARY-BY-LOCATION,REPORT=ROADLOG,DATA=INTERSTATE
/*
```

SUMMARY-BY-LOCATION accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
TABLES	Tables
PRINTER	Printed output (this name can be changed by using a PRINTER-DD or OUTPUT-FILE parameter)
SYSPRINT	Used for IBM error messages

The FORHWY-SUMMARY Program

FORHWY-SUMMARY prints either of two summaries of forest highway mileage. One of the summary formats consists of a summary by route number and location. The other format is a summary by route number and surface type.

To use the program, prepare a command that specifies the FORHWY-SUMMARY program. Include the parameter REPORT=ROADLOG and a FHSUMMARY parameter:

$$\text{FHSUMMARY=}\left\{\begin{matrix}\text{LOCATION}\\ \text{SURF-TYPE}\end{matrix}\right\}$$

You may also include any of the print parameters shown in chapter 1.

Figure 5-7 shows a sample run of the location summary. Figure 5-8 shows the second of two pages of the surface type summary. The job setup was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=LOCATION
:FORHWY-SUMMARY,REPORT=ROADLOG,FHSUMMARY=SURF-TYPE
/*
```

FORHWY-SUMMARY accesses the same files as shown above for SUMMARY-BY-LOCATION.

MONTANA FOREST HIGHWAY SYSTEM

CONSTRUCTION DETAILS

FOREST HIGHWAY ROUTE	ROUTE LENGTH PRIMARY	ROUTE LENGTH SECONDARY	IMPROVEMENT STATUS CONSTR	IMPROVEMENT STATUS UNIMP	IMPROVEMENT STATUS WYE	LOCATED IN CITY	LOCATED IN COUNTY	LOCATED IN NATIONAL FOREST	LOCATED IN STATE FOREST	LOCATED IN INDIAN RESERVE	LOCATED IN PARKS	GRAND TOTAL
1	32.378		32.378			1.816	11.291	19.271				32.378
2	20.560		20.560				7.927	12.633				20.560
3	7.411		7.411				5.307	2.104				7.411
5		34.620	34.620				11.595	23.025				34.620
6	68.686		68.686			1.052	52.100	15.534				68.686
7		22.926	13.458	9.468			3.526	19.400				22.926
8	85.454		85.454			3.455	58.957	23.042				85.454
9		21.467	21.465				2.552	13.913				21.465
10	5.094		5.094					5.094				5.094
11		32.533	32.533				16.552	15.981				32.533
12	21.961		21.961				.584	11.555	9.822			21.961
13	68.205		68.205					61.749		2.824	3.632	68.205
14	5.585		5.585					5.585				5.585
15		63.044	63.044				12.644	50.400				63.044
16	31.601		31.601					31.601				31.601
17	21.459		21.459			.231	8.636	12.592				21.459
18		7.602	7.602				7.602					7.602
19	23.563		23.563				5.989	17.574				23.563
20	16.117		16.098				2.242	13.856				16.098
21		7.960	7.960				5.342	2.618				7.960
23		5.384	5.384				5.384					5.384
24	8.808		8.808				.147	8.661				8.808
25		4.400		4.400				4.400				4.400
26	11.826		11.826					11.826				11.826
28	10.928		10.928				6.588	4.340				10.928
29		2.400		2.400				2.400				2.400
31	9.443		9.443				.142	9.301				9.443
32	52.756		52.756			3.152	10.689	38.915				52.756
40		8.765	5.166	3.599			3.566	5.199				8.765
42	41.948		41.948				3.946	38.002				41.948
43	16.585		16.585				2.049	14.536				16.585
45	8.815	22.440	31.247			.553	1.586	29.108				31.247
46	9.293		9.293			.673		8.620				9.293
51		14.949	9.949	5.000			5.232	9.717				14.949
52		13.502	5.260	8.240			2.500	11.000				13.500
53	22.922		22.922				8.022	14.900				22.922
55		2.711	2.711					2.711				2.711
56		42.702		42.700			28.130	4.570	10.000			42.700
57	61.554		61.541			.645	17.128	43.768				61.541
59	32.411		32.411			.227	4.234	27.950				32.411
60	8.564		8.564				8.564					8.564
61		57.622	11.742	45.878		1.189	8.445	47.986				57.620
PRIM	703.927		703.891			11.904	214.442	461.267	9.822	2.824	3.632	703.891
SEC		365.042	243.330	121.685		1.189	114.656	239.170	10.000			365.015
TOTAL	703.927	365.042	947.221	121.685		13.093	329.098	700.437	19.822	2.824	3.632	1068.906

Figure 5-7. FORHWY-SUMMARY Program - FHSUMMARY=LOCATION.

SUMMARY OF SURFACE TYPES -- FOREST HIGHWAY SYSTEM

NET CONSTRUCTED LENGTH -- BY FOREST HIGHWAY ROUTES

FOREST RT.NO.	SYSTEM	PRIM-ITIVE	UNIM-PROVED	GRADED & DRAINED	GRAVEL	BIT SURF TREATED	ROAD MIX	PLANT MIX	P. C. CONCRETE	TOTAL
28	PRIMARY						3.486	7.442		10.928
29	SECONDARY					2.400				2.400
31	PRIMARY						9.443			9.443
32	PRIMARY					5.891	2.079	44.786		52.756
40	SECONDARY		3.599		2.465			2.701		8.765
42	PRIMARY							41.948		41.948
43	PRIMARY					5.766		10.819		16.585
45	PRIMARY						.311	8.500		8.811
	SECONDARY							22.436		22.436
	TOTAL						.311	30.936		31.247
46	PRIMARY						1.244	8.049		9.293
51	SECONDARY				5.000	9.949				14.949
52	SECONDARY			5.540	7.960					13.500
53	PRIMARY							22.922		22.922
55	SECONDARY							2.711		2.711
56	SECONDARY		2.900		35.530	4.270				42.700
57	PRIMARY					5.629	3.454	52.458		61.541
59	PRIMARY					8.705	4.461	19.247		32.411
60	PRIMARY					8.564				8.564
61	SECONDARY			7.500	37.753	3.358		9.009		57.620
TOTALS	PRIMARY			1.155	24.222	72.276	150.854	454.424	.960	703.891
	SECONDARY		15.917	13.040	102.024	19.977	13.352	200.705		365.015
	TOTAL		15.917	14.195	126.246	92.253	164.206	655.129	.960	1068.906

Figure 5-8. FORHWY-SUMMARY Program - FHSUMMARY=SURF-TYPE.

The STATE-MILEAGE-5Ø2 Program

STATE-MILEAGE-5Ø2 prints a breakdown a mileage by surface type and by location. The location headings are:

Rural - includes all mileage outside the city limits of any incorporated cities.

Mun.under - includes all mileage within the city limits of any incorporated cities with population 4,999 or less.

Mun.over - includes all mileage within the city limits of any incorporated cities with population 5,ØØØ or over.

Urban - includes all mileage within the city or urban limits of any urban area (includes all of mun.over mileage plus some rural mileage).

To use the program, prepare a command that specifies the STATE-MILEAGE-5Ø2 program. Include a DATA parameter (see chapter 1). Include any of the following parameters as needed (these are described earlier in this chapter in the section "Commands - Roadlog Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
RLG-TYPES
SELECT-DD and/or SELECT-SIZE
MAX-#-ENTRIES

You may also include any of the print parameters shown in chapter 1.

Figure 5-9 shows a sample output from STATE-MILEAGE-5Ø2. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:STATE-MILEAGE-5Ø2,DATA=PRIMARY
/*
```

STATE-MILEAGE-5Ø2 accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
TRAFFIC	Traffic file - if referred to in a select statement
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	Used for IBM error messages

SUMMARY BY SURFACE TYPE *-- FEDERAL AID PRIMARY SYSTEM

	RURAL	MUN.UNDER	URBAN	TOTAL	MUN.OVER
A	11.797	.164		11.961	
B					
C	8.231			8.231	
E	27.465	.053		27.523	
F,G-1,H-1	987.685	13.791	4.810	1003.880	2.404
G-2,H-2,I,K,L,M	3878.824	75.785	78.065	3998.277	43.668
J	1.143	1.653	3.832	5.762	2.966
TOTAL	4915.145	91.451	86.707	5055.634	49.038

Figure 5-9. STATE-MILEAGE-5Ø2 Program.

The STATE-MILEAGE-5Ø5 Program

STATE-MILEAGE-5Ø5 prints a set of 4 summaries broken down by surface width, surface type, and ADT. The 4 summaries are (1) rural mileage, (2) urban mileage, (3) municipal mileage, and (4) combined totals. Use of STATE-MILEAGE-5Ø5 is the same as STATE-MILEAGE-5Ø2 above.

Figure 5-1Ø on the following pages shows the combined totals portion of a STATE-MILEAGE-5Ø5 run. The run was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:STATE-MILEAGE-5Ø5,DATA=PRIMARY
/*
```

STATE-MILEAGE-5Ø5 accesses the following files:

DD Statement	File
ROADLOG	Roadlog file
TRAFFIC	Traffic file
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	Used for IBM error messages

The STATE-MILEAGE-5Ø6 Program

STATE-MILEAGE-5Ø6 prints a set of 4 summaries broken down by surface width, ADT, Number of lanes, divided/undivided code, and one-way code. The 4 summaries are the same as shown above for STATE-MILEAGE-5Ø5. Use of the program is like STATE-MILEAGE-5Ø2 above.

Figure 5-11 is the fourth summary printed by a STATE-MILEAGE-5Ø6 run. The job setup for this run was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:STATE-MILEAGE-5Ø6,DATA=PRIMARY
/*
```

STATE-MILEAGE-5Ø6 accesses the same files shown above for STATE-MILEAGE-5Ø5.

EXISTING SURFACED MILEAGE BY SURFACE TYPE, WIDTH, AND AVERAGE DAILY TRAFFIC ** SM505 **

FEDERAL AID PRIMARY SYSTEM -- ALL MILEAGE

ADT	SURFACE WIDTH: 19 AND UNDER	20-21	22-23	24-26	27-35	36-43	44-47	48 AND OVER	TOTAL
1. LOW TYPE -- D,E									
UNDER 400	23.382			.304	.840				24.526
400-999					2.997				2.997
1,000-1,999									
2,000-2,999									
3,000-3,999									
4,000-4,999									
5,000-9,999									
10,000-14,999									
15,000-19,999									
20,000-29,999									
30,000-39,999									
40,000 AND OVER									
TOTAL	23.382			.304	3.837				27.523
2. INTERMEDIATE TYPE -- F,G-1,H-1									
UNDER 400	15.098	146.101	53.055	175.064	43.193	.602	.293	1.293	434.699
400-999	17.821	178.467	64.199	82.287	43.089	.011		.323	386.197
1,000-1,999	10.599	46.594	52.141	40.202	6.035	3.606			159.177
2,000-2,999	.420	5.351	.390	6.143	7.654				19.958
3,000-3,999		3.237		.558	.024				3.819
4,000-4,999						.030			.030
5,000-9,999									
10,000-14,999									
15,000-19,999									
20,000-29,999									
30,000-39,999									
40,000 AND OVER									
TOTAL	43.938	379.750	169.785	304.254	99.995	4.249	.293	1.616	1003.880
3. HIGH TYPE -- G-2,H-2,I,J,K,L									
UNDER 400	5.178	166.494	127.232	354.253	371.236	28.845	5.281	39.241	1097.810
400-999	4.174	235.214	121.388	389.249	519.891	119.334	7.960	5.190	1402.400
1,000-1,999	6.463	106.619	100.084	409.350	373.193	123.516	8.007	11.322	1138.554
2,000-2,999	1.400	6.777	9.803	58.919	30.983	34.535	7.385	6.513	156.315
3,000-3,999	.004	12.046	7.595	30.669	32.103	46.446	3.561	10.467	142.891
4,000-4,999				8.354	11.015	7.020	.492	2.352	29.233
5,000-9,999		.720		12.287	1.539	4.883	.916	14.764	35.109
10,000-14,999							.162	1.241	1.403
15,000-19,999								.324	.324
20,000-29,999									
30,000-39,999									
40,000 AND OVER									
TOTAL	17.219	527.870	366.102	1263.081	1339.960	364.629	33.764	91.414	4004.039

Figure 5-1Ø. STATE-MILEAGE-5Ø5 Program. Page 1 of 2.

EXISTING SURFACED MILEAGE BY SURFACE TYPE, WIDTH, AND AVERAGE DAILY TRAFFIC ** SM505 **

FEDERAL AID PRIMARY SYSTEM -- ALL MILEAGE

ADT	19 AND UNDER	20-21	22-23	24-26	27-35	36-43	44-47	48 AND OVER	TOTAL
	**************************** SURFACE WIDTH							******************************	
	---------------- 4. COMBINED TOTALS ----------------								
UNDER 400	43.658	312.595	180.287	529.621	415.269	29.497	5.574	40.534	1557.035
400-999	21.995	413.681	185.587	471.536	565.977	119.345	7.960	5.513	1791.594
1,000-1,999	17.062	153.213	152.225	449.552	379.228	127.122	6.607	11.322	1297.731
2,000-2,999	1.820	12.128	10.193	65.062	38.637	34.535	7.385	6.513	176.273
3,000-3,999	.004	15.283	7.595	31.227	32.127	46.446	3.561	10.467	146.710
4,000-4,999				8.354	11.015	7.050	.492	2.352	29.263
5,000-9,999		.720		12.267	1.539	4.883	.916	14.764	35.109
10,000-14,999							.162	1.241	1.403
15,000-19,999								.324	.324
20,000-29,999									
30,000-39,999									
40,000 AND OVER									
TOTAL	84.539	907.620	535.887	1567.639	1443.792	368.878	34.057	93.030	5035.442

Figure 5-1Ø. STATE-MILEAGE-5Ø5 Program. Page 2 of 2.

EXISTING MILEAGE OF SURFACED HIGHWAYS BY TRAFFIC LANES AND ACCESS CONTROL **SM506**

FEDERAL AID PRIMARY SYSTEM -- ALL MILEAGE

ADT	31 AND UNDER	32-37	38-43	44-47	48 AND OVER	TOTAL
	****** SURFACE WIDTH ******					
1. ONE WAY STREETS						
UNDER 400						
400-999						
1,000-1,999						
2,000-2,999						
3,000-3,999						
4,000-4,999						
5,000-9,999						
10,000-14,999						
15,000-19,999						
20,000-29,999						
30,000-39,999						
40,000 AND OVER						
TOTAL						
2. UNDIVIDED 4 OR MORE LANES						
UNDER 400			4.177	.480	16.469	21.126
400-999			1.326	4.121	3.149	8.596
1,000-1,999			.898		9.125	10.023
2,000-2,999			.698		3.610	4.308
3,000-3,999			1.235		3.275	4.510
4,000-4,999					.825	.825
5,000-9,999				.237	2.013	2.250
10,000-14,999				.162	.881	1.043
15,000-19,999						
20,000-29,999						
30,000-39,999						
40,000 AND OVER						
TOTAL			8.334	5.000	39.347	52.681
3. DIVIDED 4 OR MORE LANES NO CONT						
UNDER 400				.220	12.064	12.284
400-999						
1,000-1,999				.453	.377	.830
2,000-2,999				2.550	2.903	5.453
3,000-3,999			.048	.024	4.319	4.391
4,000-4,999			.882		1.527	2.409
5,000-9,999					12.666	12.666
10,000-14,999					.360	.360
15,000-19,999					.324	.324
20,000-29,999						
30,000-39,999						
40,000 AND OVER						
TOTAL			.930	3.247	34.540	38.717

Figure 5-11. STATE-MILEAGE-5Ø6 Program. Page 1 of 2.

EXISTING MILEAGE OF SURFACED HIGHWAYS BY TRAFFIC LANES AND ACCESS CONTROL **SM506**

FEDERAL AID PRIMARY SYSTEM -- ALL MILEAGE

****************************** SURFACE WIDTH ******************************

---------------- 4. DIVIDED 4 OR MORE LANES PARTIAL

ADT	31 AND UNDER	32-37	38-43	44-47	48 AND OVER	TOTAL
UNDER 400					.163	.163
400-999						
1,000-1,999						
2,000-2,999						
3,000-3,999						
4,000-4,999						
5,000-9,999						
10,000-14,999						
15,000-19,999						
20,000-29,999						
30,000-39,999						
40,000 AND OVER						
TOTAL					.163	.163

---------------- 5. DIVIDED 4 OR MORE LANES FULL CONT

ADT	31 AND UNDER	32-37	38-43	44-47	48 AND OVER	TOTAL
UNDER 400						
400-999						
1,000-1,999						
2,000-2,999						
3,000-3,999						
4,000-4,999						
5,000-9,999						
10,000-14,999						
15,000-19,999						
20,000-29,999						
30,000-39,999						
40,000 AND OVER						
TOTAL						

---------------- 6. COMBINED TOTALS

ADT	31 AND UNDER	32-37	38-43	44-47	48 AND OVER	TOTAL
UNDER 400			4.177	.700	28.696	33.573
400-999			1.325	4.121	3.149	8.596
1,000-1,999			.898	.453	9.502	10.853
2,000-2,999			.698	2.550	6.513	9.761
3,000-3,999			1.283	.024	7.594	8.901
4,000-4,999			.382		2.352	3.234
5,000-9,999				.237	14.679	14.916
10,000-14,999				.162	1.241	1.403
15,000-19,999					.324	.324
20,000-29,999						
30,000-39,999						
40,000 AND OVER						
TOTAL			9.264	8.247	74.050	91.561

Figure 5-11. STATE-MILEAGE-5Ø6 Program. Page 2 of 2.

The LOCAL-BY-ROAD# Program

LOCAL-BY-ROAD# is designed for use with local system data. It prints a listing of local records plus a summary of section lengths by route number and by county. The data elements printed are:

County number
Route number
Milepoint
Remark
Location code
Functional classification
City number
Year
Surface type
Surface and roadway width (feet)
Section length (miles)
Section, township, and range
Jurisdiction
Area name

To use the program, prepare a command that specifies the LOCAL-BY-ROAD# program. Include the parameter FILE=ROADLOG and the parameter:

DATA=LOCAL [=n [-n]]

You may also include any of the following parameters (these are described earlier in this chapter in the section "Commands - Roadlog Subsystem"):

START-MILEPOINT and/or END-MILEPOINT
RLG-TYPES
SELECT-DD and/or SELECT-SIZE
MAX-#-ENTRIES

You may also include any of the print parameters shown in chapter 1.

Figure 5-12 shows one page of a LOCAL-BY-ROAD# output. The job setup for this run was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:LOCAL-BY-ROAD#,FILE=ROADLOG,DATA=LOCAL=Ø1ØØ1-Ø1Ø12
/*
```

BEAVERHEAD COUNTY

LOCAL ROAD - RURAL - INVENTORY LISTING

04/07/76

PAGE 1

CNTY NO	ROAD NO	MILE POST	RMRK	CNTY NO	AREA SERV	FUNCT CLASS	CITY	YR	SURF TYPE	SURF WDTH	ROAD WDTH	SECT LNGTH	LOCATION SECT	TOWN	RNGE	JURIS	AREA NAME
01	001	000+0.000		01	05	6	000	74	0002	16	16	00.970	03	02S	19W	02	65
							ROAD NUMBER 001 -					0.970 MILES					
01	002	000+0.000		01	05	6	000	74	2010	16	16	06.160	05	02S	18W	02	65
01	002	006+0.160		01	05	6	000	74	2010	16	16	00.020	23	01S	19W	02	65
01	002	006+0.180		01	05	6	000	74	0002	16	16	00.820	23	01S	19W	02	65
							ROAD NUMBER 002 -					5.020 MILES					
01	003	000+0.000		01	05	6	000	74	0002	16	16	05.380	34	01S	19W	02	65
							ROAD NUMBER 003 -					5.380 MILES					
01	004	000+0.000		01	05	6	000	74	0010	16	16	00.800	24	02S	18W	02	65
							ROAD NUMBER 004 -					0.800 MILES					
01	005	000+0.000		01	05	6	000	74	2010	16	16	06.840	16	02S	17W	02	65
01	005	006+0.840		01	05	6	000	74	0002	12	12	05.080	04	02S	17W	02	65
							ROAD NUMBER 005 -					11.920 MILES					
01	006	000+0.000		01	05	6	000	74	2010	18	18	11.450	27	02S	17W	02	65
							ROAD NUMBER 006 -					11.450 MILES					
01	009	000+0.000		01	06	6	000	74	4231	24	24	00.770	24	02S	17W	15	00
							ROAD NUMBER 009 -					0.770 MILES					
01	010	000+0.000		01	03	5	000	74	2010	16	18	07.130	25	02S	16W	01	00
01	010	007+0.130		01	03	5	000	74	2010	18	19	00.060	34	01S	16W	01	00
01	010	007+0.190		01	03	5	000	74	2010	18	18	00.260	34	01S	16W	01	00
01	010	007+0.450		01	03	5	000	74	2010	18	18	02.250	34	01S	16W	01	10
01	010	009+0.700		01	03	5	000	74	2010	20	20	05.740	22	01S	16W	01	00
							ROAD NUMBER 010 -					15.440 MILES					
01	011	000+0.000		01	03	6	000	74	0002	12	12	01.420	03	02S	16W	01	00
							ROAD NUMBER 011 -					1.420 MILES					
01	012	000+0.000		01	03	6	000	74	2010	18	18	03.930	34	01S	16W	01	00

Figure 5-12. LOCAL-BY-ROAD# Program.

The LOCAL-BY-SURF-TYPE Program

LOCAL-BY-SURF-TYPE is designed for use with local system data. It produces a summary of mileage broken down by jurisdiction, location code, and surface type for each county plus a summary of all counties. Use of the program is identical to LOCAL-BY-ROAD#.

Figure 5-13 shows the summary of all counties portion of a LOCAL-BY-SURF-TYPE run. The job setup was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISRLG
//SYSIN DD *
:LOCAL-BY-SURF-TYPE,FILE=ROADLOG,DATA=LOCAL=Ø1ØØ1-Ø1Ø12
/*
```

THE SUMMARY OF RAODLOG SECTION LENGTHS FOR BEAVERHEAD COUNTY

BY JURISDICTION:

TYPE	AMOUNT
COUNTY	20.790
NATIONAL FOREST	39.540
INDIAN RESERVATION	0.000
NATIONAL PARK	0.000
NATIONAL MONUMENT	0.770
NATIONAL WILDLIFE REFUGE	0.000
BUREAU OF LAND MANAGEMENT	0.000
STATE PARK	0.000
STATE FOREST	0.000
STATE GAME PRESERVE	0.000
TOTAL	61.100

BY SURFACE TYPE:

CODE	TYPE	AMOUNT
0001	PRIMITIVE	0.000
0002	UNIMPROVED	16.670
0010	GRADED AND DRAINED	0.800
2010	GRAVEL	42.860
3210	BITUM SURF TREAT	0.000
4231	ROAD MIX	0.770
7201	P. C. CONCRETE	0.000
	TOTAL	61.100

BY LOCATION:

TYPE	AMOUNT
CITY	0.000
URBAN	0.000
COUNTY	20.790
INDIAN RESERVATION	0.000
NATIONAL FOREST	39.540
NATIONAL MONUMENT	0.770
NATIONAL PARK	0.000
NATIONAL WILDLIFE REFUGE	0.000
MILITARY RESERVATION	0.000
STATE FOREST	0.000
STATE PARK	0.000
STATE GAME PRESERVE	0.000
BUREAU OF LAND MANAGEMENT	0.000
FRONTAGE	0.000
PRIVATE	0.000
TOTAL	61.100

Figure 5-13. LOCAL-BY-SURF-TYPE Program.

CHAPTER 6

THE SKID SUBSYSTEM

Introduction

The skid file contains the results of skid tests performed on roadways. Each entry ("record") in the file contains the results of one test. Each record contains the following data elements:

Route system and route number
Milepoint (rural tests only)
Direction test vehicle traveled
Lane
Wheel (left or right) used in test
Date of test
City number and x- and y-coordinates (municipal tests only)
Speed of test vehicle
Observed skid number
Surface type
Surface texture
Curvature
Grade
Pavement temperature
Foreign matter on roadway
Location
Surface repair
Comments

Standard User Input - Skid Subsystem

Because the Highway Information System has been designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the skid programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISSKID
//SYSIN DD *
   One or more commands
/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed by the computer.

The job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control

information may be necessary. When submitting a run from a CRJE terminal you can direct the output to the terminal. A sample job setup is:

```
// JOB
// EXEC HISSKID,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *
   One or more commands
/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PARM='PRINTER-DD=CRJE' instead of PARM='OUTPUT-FILE=CRJE'.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Skid Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section describes parameters that can be used on several of the skid commands.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters are used to select rural skid tests by route system, route number, and milepoint. They are described in detail in chapter 1.

The MAINT-DIV Parameter - MAINT-DIV can be used to select rural skid tests based on maintenance division. Code the parameter as MAINT-DIV=nn, where "nn" is one of the maintenance divisions. You may also use the format MAINT-DIV=ALL to request all of the maintenance divisions. MAINT-DIV and DATA can be used in conjunction as in MAINT-DIV=21,DATA=INTERSTATE (selects tests on interstate highways in maintenance division 21).

The CITY Parameter - The CITY parameter is used to select municipal skid tests. Code CITY=ALL to select all of the municipal tests. Code CITY=city to select all of the municipal tests in the specified city (use the city names shown in table 2-1 in chapter 2).

The START-DATE and END-DATE Parameters - These parameters can be used to select skid tests based on date. They are described in chapter 2 of this manual.

The DIRECTION Parameter - DIRECTION can be used to select skid tests based on the contents of the direction field. The format of the parameter is:

$$\text{DIRECTION=}\left\{\begin{array}{l}\text{INCREASING}\\ \text{DECREASING}\\ \text{RAMP}\\ \text{NORTH}\\ \text{SOUTH}\\ \text{EAST}\\ \text{WEST}\\ \text{n}\end{array}\right\}$$

"n" can range from Ø to 9.

The LANE Parameter - LANE can be used to select skid tests based on the contents of the lane field. The format of the parameter is:

$$\text{LANE=}\left\{\begin{array}{l}\text{INSIDE}\\ \text{MIDDLE}\\ \text{OUTSIDE}\\ \text{LEFT-TURN}\\ \text{RIGHT-TURN}\\ \text{ACCEL}\end{array}\right\}$$

The WHEEL Parameter - WHEEL can be used to select skid tests based on the contents of the wheel field. The format of the parameter is:

$$\text{WHEEL=}\left\{\begin{array}{l}\text{LEFT}\\ \text{RIGHT}\end{array}\right\}$$

The EFF-DATE Parameter - EFF-DATE can be used to select rural tests based on the roadlog effective date. EFF-DATE=NO, the default, does not compare the skid tests against the roadlog file. EFF-DATE compares the date of each skid test against the roadlog effective date and skips any tests performed before the effective date.

The X-COORDINATE and Y-COORDINATE Parameters - These parameters, used in conjunction with SQUARE-SIZE, can be used to select municipal skid tests that occurred within a square area of a city. A CITY parameter must also be coded on the command and must specify a city name. Use X-COORDINATE and Y-COORDINATE to identify the center point of the square. Use SQUARE-SIZE to specify the size of the square.

The SQUARE-SIZE Parameter - SQUARE-SIZE must be used in conjunction with X-COORDINATE and Y-COORDINATE. Specify the length (in coordinate units) of a side of the square.

The MAX-#-ENTRIES Parameter - MAX-#-ENTRIES can be used to limit the number of skid tests processed. Only those tests that meet all of the specified selection criteria are counted.

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting skid tests. The SELECT-DD parameter provides the capability of choosing skid tests that meet selection criteria based on the contents of nearly any data element(s) in the skid, roadlog, and traffic files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on the use of SELECT-DD and SELECT-SIZE is included in chapter 12. Do not attempt to select based on the roadlog or traffic file when you specify a CITY parameter on the command.

The LIST Program

The LIST program prints the results of skid tests from the skid file. One line is printed for each skid test. The program can run in either of two modes: (1) "select" mode wherein skid tests can be selected based on command parameters, and (2) "specific" mode wherein specific tests identified by key are printed.

To use the program, prepare a command that specifies the program LIST. Include the parameter FILE=SKID. You may include the following parameter:

$$\text{CODE-LIST=} \begin{Bmatrix} \text{NO} \\ \underline{\text{YES}} \end{Bmatrix}$$

CODE-LIST=YES (the default) prints a list of the codes that appear in the listing. CODE-LIST=NO suppresses the code listing (this option is useful when obtaining output over a CRJE terminal).

To run in select mode, include one or more of the parameters DATA, MAINT-DIV, and CITY. DATA and/or MAINT-DIV selects rural tests based on route or on maintenance division or both. CITY selects municipal tests (either all tests or just those in a specific city). These parameters are described earlier in this chapter in the section "Commands- Skid Subsystem". You may also include any of the following parameters:

START-MILEPOINT and/or END-MILEPOINT (only if DATA is included)
START-DATE and/or END-DATE
DIRECTION
LANE
WHEEL
EFF-DATE (only if DATA and/or MAINT-DIV is included)
SQUARE-SIZE, X-COORDINATE, and Y-COORDINATE (only if CITY is included)
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

You may also include any of the print parameters described in chapter 1.

Figure 6-1 shows a sample output from LIST. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSKID
//SYSIN DD *
:LIST,FILE=SKID,CODE-LIST=NO,DATA=PRIMARY=7,END-MILEPOINT=Ø2Ø+Ø.ØØØ
/*
```

```
        SKID LISTING -- MILEPOINT LOCATIONS                    APR 7,1976

ROUTE                D L W                      SK S S H G PAV F L S                                                  DATE OF
NUMBER MILEPOINT     R V L  --DATE-- SEQ. SP NO T X C D TEM M C R ---------------COMMENTS---------------  UPDATE

P00007 001+0.000     D 0 L  7/00/73 0001 40 30                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 001+0.000     I 0 L  7/00/73 0001 40 23                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 002+0.000     D 0 L  7/00/73 0001 40 22                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 002+0.000     I 0 L  7/00/73 0001 40 16                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 003+0.000     D 0 L  7/00/73 0001 40 32                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 003+0.000     I 0 L  7/00/73 0001 40 28                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 004+0.000     D 0 L  7/00/73 0001 40 24                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 004+0.000     I 0 L  7/00/73 0001 42 25                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 005+0.000     D 0 L  7/00/73 0001 40 38                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 005+0.000     I 0 L  7/00/73 0001 42 33                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 005+0.000     D 0 L  7/00/73 0001 40 30                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 006+0.000     I 0 L  7/00/73 0001 40 29                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 007+0.000     I 0 L  7/00/73 0001 40 22                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 008+0.000     D 0 L  7/00/73 0001 40 42                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 008+0.000     I 0 L  7/00/73 0001 40 39                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 008+0.705     D 0 L  7/00/73 0001 40 27                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 009+0.533     D 0 L  7/00/73 0001 40 40                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 010+0.000     I 0 L  7/00/73 0001 40 47                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 010+0.722     D 0 L  7/00/73 0001 40 24                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 011+0.000     I 0 L  7/00/73 0001 40 18                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 012+0.000     I 0 L  7/00/73 0001 40 19                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 012+0.715     D 0 L  7/00/73 0001 40 18                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 013+0.000     I 0 L  7/00/73 0001 40 21                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 013+0.708     D 0 L  7/00/73 0001 40 27                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 014+0.000     I 0 L  7/00/73 0001 40 28                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 014+0.700     D 0 L  7/00/73 0001 40 23                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 015+0.000     I 0 L  7/00/73 0001 40 21                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 015+0.373     D 0 L  7/00/73 0001 40 30                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 016+0.000     I 0 L  7/00/73 0001 40 23                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 016+0.522     D 0 L  7/00/73 0001 40 20                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 017+0.000     D 0 L  7/00/73 0001 40 54                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 017+0.000     I 0 L  7/00/73 0001 42 45                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 018+0.000     D 0 L  7/00/73 0001 40 57                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=S
P00007 018+0.000     I 0 L  7/00/73 0001 40 52                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
P00007 019+0.000     I 0 L  7/00/73 0001 40 46                   OLD-TEST-1= ,OLD-TEST-2= ,OLD-DIREC=N
```

Figure 6-1. Skid LIST Program.

To use the LIST program in specific mode, prepare a command that specifies the LIST program and include the parameter FILE=SKID. You may include CODE-LIST as shown above. Include a DDNAME parameter that specifies the name of a DD statement used to enter actual keys. You may also include any of the print parameters shown in chapter 1. Submit the job in the format shown:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSKID
//SYSIN DD *
:LIST,FILE=SKID,DDNAME=SKIDLIST
/*
//SKIDLIST DD *
   One or more data cards
/*
```

You may substitute any name that does not occur in HISSKID for "SKIDLIST." On each data card code the complete key (28 characters) of a skid test. The skid key format is provided in Appendix B of this manual.

The skid LIST program accesses the following files:

DD Statement	File
SKID	Skid file
ROADLOG	Roadlog file - if referred to in a select statement or if an EFF-DATE=YES parameter is included
TRAFFIC	Traffic file - if referred to in a select statement
TABLES	Tables - if a select statement is included
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	Used for IBM error messages
ddname	Input data cards - if DDNAME is specified

The LOW-SKID-NUMBERS Program

LOW-SKID-NUMBERS prints a report of locations having low skid numbers (4Ø or less). To use the program, prepare a command that specifies the LOW-SKID-NUMBERS program. Include one or more of the parameters DATA, MAINT-DIV, and CITY. Also include any of the following parameters as needed (these parameters are described earlier in this chapter in the section "Commands - Skid Subsystem"):

START-MILEPOINT and/or END-MILEPOINT (only if DATA included)
START-DATE and/or END-DATE
DIRECTION
LANE
WHEEL
EFF-DATE (only if DATA and/or MAINT-DIV is included)
SQUARE-SIZE, X-COORDINATE, and Y-COORDINATE (only if CITY=city included)
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

You may also include any of the print parameters described in chapter 1.

Figure 6-2 shows a sample output from LOW-SKID-NUMBERS. This run was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSKID
//SYSIN DD *
:LOW-SKID-NUMBERS,DATA=PRIMARY=7,END-MILEPOINT=Ø2Ø+Ø.ØØØ
/*
```

LOW-SKID-NUMBERS accesses the following files:

DD Statement	File
SKID	Skid file
ROADLOG	Roadlog file - if referred to in a select statement or if EFF-DATE is specified as EFF-DATE=YES
TRAFFIC	Traffic file - if referred to in a select statement
TABLES	Tables - if a select statement is included
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	Used for IBM error messages

LOW-SKID-NUMBERS STATEWIDE APR 7,1976

--------- SKID NUMBERS BELOW 20 ---------

ROUTE NUMBER	MILEPOINT	DIR	LAN	WHL	SPEED	SKID NUMBER
P00007	002+0.000	I	0	L	40	16/16
P00007	011+0.000	I	0	L	40	18/18
P00007	012+0.000	I	0	L	40	19/19
P00007	012+0.715	D	0	L	40	18/18

----- SKID NUMBERS BETWEEN 20 AND 30 ----

ROUTE NUMBER	MILEPOINT	DIR	LAN	WHL	SPEED	SKID NUMBER
P00007	001+0.000	I	0	L	40	23/23
P00007	002+0.000	D	0	L	40	22/22
P00007	003+0.000	I	0	L	40	28/28
P00007	004+0.000	D	0	L	40	24/24
P00007	004+0.000	I	0	L	42	**/25
P00007	006+0.000	I	0	L	40	29/29
P00007	007+0.000	I	0	L	40	22/22
P00007	008+0.705	D	0	L	40	27/27
P00007	010+0.722	D	0	L	40	24/24
P00007	013+0.000	I	0	L	40	21/21
P00007	013+0.708	D	0	L	40	27/27
P00007	014+0.000	I	0	L	40	28/28
P00007	014+0.700	D	0	L	40	23/23
P00007	015+0.000	I	0	L	40	21/21
P00007	016+0.000	I	0	L	40	23/23
P00007	016+0.522	D	0	L	40	20/20

----- SKID NUMBERS BETWEEN 30 AND 40 ----

ROUTE NUMBER	MILEPOINT	DIR	LAN	WHL	SPEED	SKID NUMBER
P00007	003+0.000	D	0	L	40	32/32
P00007	005+0.000	D	0	L	40	38/38
P00007	005+0.000	I	0	L	42	**/33
P00007	006+0.000	D	0	L	40	30/30
P00007	008+0.000	I	0	L	40	39/39
P00007	009+0.683	D	0	L	40	40/40
P00007	015+0.873	D	0	L	40	30/30

Figure 6-2. LOW-SKID-NUMBERS Program.

CHAPTER 7

THE SUFFICIENCY SUBSYSTEM

Introduction

The Montana Department of Highway's Planning and Research Bureau has adopted a sufficiency rating system in order to compare the sufficiency of existing rural highways with the latest design standards. This rating system takes into consideration the structural adequacy, the safety, and the traffic capacity of the highway.

The sufficiency file at present contains rating information only for Montana's Federal Aid Primary route system. The file has been designed, however, to store sufficiency rating information for other systems as the data becomes available.

Each highway is broken down into a number of "sections" for sufficiency reporting purposes. A sufficiency section is identified by specifying the beginning milepoint of the section. Section breaks begin at any point along the highway where a significant change in the highway's structural adequacy, safety, or traffic capacity occurs as well as at points of jurisdictional change (city limits, county lines, etc.).

The sufficiency file contains one entry (or "record") for each sufficiency section. Each record contains the following data elements:

Route number and route system
Milepoint at beginning of section
Verbal description
Design speed
Terrain
Average speed
Percent of sight distance less than design
Number of stopping sight distances less than design
Number of curves sharper than design degree of curvature
Number of narrow bridges
Foundation rating
Surface rating
Drainage rating
Section length
Effective date

When calculating sufficiency ratings, data is also gathered from the accident, roadlog, traffic, and true mileage files.

The Sufficiency Report File

The annual sufficiency report requires that data be gathered from a number of files and that a number of computations be performed on the data. A number of reports and summaries must be generated from the gathered and computed data. The system has been designed such that the data gathering and computations are performed just once and the results stored in a subsidiary file, the "sufficiency report file." The sufficiency report file must be generated by system maintenance personnel before you can use most of the programs described in this chapter.

The sufficiency report file contains the following data elements:

Route system and route number (from sufficiency file)
Milepoint at beginning of section (from sufficiency file)
Verbal description (from sufficiency file)
County number (from roadlog file)
Financial district (computed from county number)
Year built and year improved (from roadlog file)
Surface width and roadway width (from roadlog file)
Surface type (from roadlog file)
Section length (computed from true mileage file)
Average daily traffic (computed from traffic file)
Design hour volume (from traffic file)
Percent of commercial traffic (from traffic file)
Service volume (computed)
Number of accidents (from accident files)
Foundation rating (from sufficiency file)
Surface rating (from sufficiency file)
Drainage rating (from sufficiency file)
Safety rating (computed)
Capacity rating (computed)
Total rating (computed)
Adjusted rating (computed)
Deficient mileage (computed)
Design speed (from sufficiency file)
Terrain (from sufficiency file)
Average speed (from sufficiency file)
Sight distance (from sufficiency file)
Stopping distance (from sufficiency file)
Number of curves (from sufficiency file)
Number of narrow bridges (from sufficiency file)
Number of lanes (from roadlog file)
Divided/undivided code (from roadlog file)
City number (from roadlog file)
Current year traffic (computed from traffic file)

Computation of Service Volume

For 4-lane roads:

Service volume = 8ØØØxy.

$$x = \frac{100}{100 + a(2^{b}-1)}$$

a = percent of commercial traffic
b = terrain code
y = factor extracted from tables utilizing speed factors and width factors. The speed factor is determined from the average speed and the design speed. The width factor is determined from the shoulder width. y is the product of the speed factor and the width factor.

For 2- and 3-lane roads:

Service volume = xy.

(2-lane) x = adjustment factor extracted using terrain code, percent commercial, lane width, and shoulder width.

(3-lane) x = lane factor extracted using lane width, shoulder width, and terrain.

y = Hourly volume factor.

NOTE: The sufficiency tables (speed factors, adjustment factors, lane factors, hourly volume factors, and width factors) are described in the publication Highway Information System Release 4.Ø - Record Formats and Subroutines.

Computation of Capacity Rating

$$\text{Capacity rating} = 30 - \frac{15a}{b}$$

a = design hourly volume
b = service volume

Computation of Safety Rating

$$\text{Safety rating} = \frac{2x}{N_1+N_2+N_3+N_4}$$

x = section length

N_1 = Number of stopping sight distances less than permitted by design

N_2 = Number of curves sharper than design degree of curvature

N_3 = Number of narrow bridges

N_4 = Accident rate

$$\text{Accident rate} = \frac{N \times 10^7}{\text{AADT} \times 365 \times 100}$$

N = number of accidents
AADT = 3-year annual average daily traffic

Computation of Total Rating

Total rating = foundation rating + surface rating + drainage rating + safety rating + capacity rating

Maximum value = 100

Computation of Deficient Mileage

$$\text{Deficient mileage} = \frac{x(100-r)}{100}$$

x = section length

r = total rating

Computation of Adjusted Rating

$$\text{Adjusted rating} = x + \frac{x^2 - 100x}{50 \log y} (\log z - \log y)$$

x = total rating
y = total system current-year AADT
z = total section current-year AADT

Standard User Input - Sufficiency Subsystem

Because the Highway Information System has been designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the sufficiency programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISSUFF
//SYSIN DD *

   One or more commands

/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed by the computer.

The job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, you can direct the output to the terminal. A sample job setup is:

```
// JOB
// EXEC HISSUFF,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *

   One or more commands

/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PARM='PRINTER-DD=CRJE' instead of PARM='OUTPUT-FILE=CRJE'.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Sufficiency Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section discusses parameters that can be coded on several of the sufficiency commands.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters provide a flexible means of selecting outputs based on route system, route number, and milepoint. They are described in detail in chapter 1 of this manual. Keep in mind that the sufficiency file only contains data for primary highways at the current time.

The MAX-#-ENTRIES Parameter - This parameter is implemented primarily as protection to the user and as an aid in program debugging. It places a limit on the number of sufficiency sections accepted on input (only sections that meet all of the requested selection criteria are counted).

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting sufficiency sections. The SELECT-DD parameter provides the capability of choosing sections that meet selection criteria based on the contents of nearly any data element(s) in the sufficiency, roadlog, and traffic files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on its use is included in chapter 12.

The LIST Program

The list program prints the contents of the sufficiency file. One line is printed for each record in the file.

To use the program, prepare a command that specifies the LIST program. Include the parameter FILE=SUFFICIENCY. Include a DATA parameter (see chapter 1). You may also include any of the following parameters as needed (these parameters are described in the preceding section):

START-MILEPOINT and/or END-MILEPOINT
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

You may also include any of the print parameters described in chapter 1.

Figure 7-1 shows a sample output from the sufficiency LIST program. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:LIST,FILE=SUFFICIENCY,DATA=PRIMARY=17
/*
```

The sufficiency LIST program accesses the following files:

DD Statement	File
SUFFICY	Sufficiency file
ROADLOG	Roadlog file - if referred to in a select statement
TRAFFIC	Traffic file - if referred to in a select statement
TABLES	Tables - if a select statement is included
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The LIST-SUFFREP Program

LIST-SUFFREP prints a dump listing of the sufficiency report file. Be sure that the report file has been created before trying to use this program.

One line is printed for each record printed. The print line has this format:

Columns	Length	Data Element
1-10	10	blank
11	1	Route system
12-16	5	Route number
17-25	9	Milepoint
26	1	Remark
27-44	18	Description
45-46	2	County number
47-48	2	Financial district
49-50	2	Year built
51-52	2	Year improved
53-54	2	Surface width
55-56	2	Roadway width
57-59	3	Surface type
60-66	7	Section length
67-71	5	3-year ADT
72-76	5	Design hour volume
77-78	2	Percent commercial
79-82	4	Service volume
83-84	2	Number of accidents
85-86	2	Foundation rating
87-88	2	Surface rating

---------- HIS SUFFICIENCY FILE LISTING ----------

-----KEY-----	---DESCRIPTION---	DESIGN SPEED	TER-RAIN	AVER SPEED	SIGHT DIST	STOP DIST	NUMBER OF CURVES	NUMBER OF NARROW BRIDGES	FOUNDATION RATING	SURF RATING	DRAIN RATING	SECTION LENGTH	EFFECTIVE DATE
P017000+0.000	MILES CITY LIMITS	60	1	65	10				2	17	10		
P017006+0.191	5.8 E MILES CITY	60	1	65						14	10		
P017008+0.358	3.0 E MILES CITY	60	2	65	60	1	1		10	25	10		
P017009+0.023	COINCIDENT	00											
P017018+0.648	DIAMOND RING INTCH	50	2	55		1	2		9	28	10		
P017019+0.177	0.6 E DIAMOND INT	50	1	60	10	1				10	5		
P017022+0.374	PRAIRIE CO LINE	60	1	65	30	5				13	6		
P017029+0.626	5.5 W TERRY	60	2	70					8	23	10		
P017031+0.365	4.0 W TERRY	60	1	65	40	5				14	6		
P017035+0.431	CITY	00											
P017036+0.554	TERRY	50	1	55						11	5		
P017037+0.851	1.3 E TERRY	60	1	65	70				6	16	9		
P017038+0.695	2.1 E TERRY	60	2	60	40	14	1			14	5		
P017043+0.815	7.2 E TERRY	60	1	65	20	20			5	16	8		
P017044+0.856	COINCIDENT	00											
P017070+0.435	END OF ROUTE	00											

HIS LIST-SUFFREP ROUTINE

```
PCO017000+0.000 MILES CITY LIMITS  91248432830RMS  6191  690  8412 631 8 217101228 69 7119196016510     2     653
PCO017006+0.191 5.8 E MILES CITY   91254542432PMS  2171  205  2712 680 1  14101129 64 73 78160165     2     212
PCO017008+0.358 3.0 E MILES CITY   91262623232FMS   659  169  2212 365  102510 129 75 84 1646026560 1 1 2     175
PCO017009+0.0230COINCIDENT                         10041
PO0017018+0.648 DIAMOND RING INTCH 91271712625PMS   579  114  2112 671   92810  30 77 85 15350255   1 2 2     168
PCO017019+0.177 0.6 E DIAMOND INT  91229412425BST  3286  114  2112 889 1  10 5 430 49 6116755016010 1   2     168
PO0017022+0.374 PRAIRIE CO LINE  40 43157222RMS  7332   87  1612 456 2  13 6 229 50 6436666016530 5   2     130
PCO017029+0.626 5.5 W TERRY      40 46662323RMS  1561  102  1512 531 2 82310 230 73 84 42160270      2     122
PCO017031+0.365 4.0 W TERRY      40 43157243ORMS  4030  164  2012 530 1  14 6 129 50 6320156016540 5   2     157
PCO017035+0.431MCITY             40 4           1002                                                 109
PCO017036+0.554 TERRY            40 431572430RMS  1294   56  1512 951     11 5203C 66 80 43950155     2     110
PCO017037+0.851 1.3 E TERRY      40 455552432RMS   844   84  1412 424    616 92030 81 91 1606016570   2     94123171
PCO017038+0.695 2.1 E TERRY      40 43149242RMS  5120   70  1212 312 1  14 5 129 49 662611602604014 1 2     85
PCO017043+0.815 7.2 E TERRY      40 449492432RMS  1041   59  1012 616 3 516 8  30 59 76 426601652020   2     80123171
PCO017044+0.856COINCIDENT                        25493                                                         123171
PCO017070+0.435END OF ROUTE
```

Figure 7-1. Sufficiency LIST and LIST-SUFFREP.

Columns	Length	Data Element
89-9∅	2	Drainage rating
91-92	2	Safety rating
93-94	2	Capacity rating
95-97	3	Total rating
98-1∅∅	3	Adjusted rating
1∅1-1∅4	4	Deficient mileage (xx.xx)
1∅5-1∅6	2	Design speed
1∅7	1	Terrain
1∅8-1∅9	2	Average speed
11∅-111	2	Number of sight distances less than design
112-113	2	Number of stopping distances less than design
114-115	2	Number of curves sharper than design
116	1	Number of narrow bridges
117	1	Number of lanes
118	1	Divided/undivided code
119-121	3	City number
122-126	5	Current-year ADT
127-132	6	Effective date

To use the program, prepare a command that specifies the LIST-SUFFREP program. Include a DATA parameter (see chapter 1). You may optionally include a START-MILEPOINT and/or an END-MILEPOINT parameter. You may also include any of the print parameters described in chapter 1.

Figure 7-1 shows a sample output of LIST-SUFFREP. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:LIST-SUFFREP,DATA=PRIMARY=17
/*
```

LIST-SUFFREP accesses the following files:

DD Statement	File
SUFFREP	Sufficiency report file
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The LIST-BY-SECTION Program

LIST-BY-SECTION prints a formatted listing of the sufficiency report file in order by route system, route number, and milepoint. Be sure the sufficiency report file has been created before attempting to use this program.

To use the program, prepare a command that specifies the LIST-BY-SECTION program. Include the parameter REPORT=SUFFICIENCY. Include a DATA parameter and, optionally, START-MILEPOINT and/or END-MILEPOINT parameters (see chapter 1). You may if you wish include any of the print parameters described in chapter 1.

Figure 7-2 shows a sample output from LIST-BY-SECTION. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:LIST-BY-SECTION,REPORT=SUFFICIENCY,DATA=PRIMARY=17
/*
```

LIST-BY-SECTION accesses the following files:

DD Statement	File
SUFFREP	Sufficiency report file
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The LIST-BY-RATING Program

LIST-BY-RATING prints a formatted listing of the sufficiency report file in order by adjusted rating. Use of the program is just like LIST-BY-SECTION.

Figure 7-2 shows a sample output from LIST-BY-RATING. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:LIST-BY-RATING,REPORT=SUFFICIENCY,DATA=PRIMARY=17
/*
```

FEDERAL AID PRIMARY ROUTE NUMBER 17

MILEPOINT	SECTION DESCRIPTION	COUNTY	FIN DIST	*YEAR* BT	*YEAR* IM	SECTION LENGTH	WIDTH SRF	WIDTH RDY	SUF TYP	ADT	DHV	SERVICE VOLUME	FND	SRF	ORN	SAF	CAP	TOT	ADJ	DEFIC MILES
0+0.000	MILES CITY LIMITS	CUSTER	12	48	48	6.191	28	30	RMS	653	84	631	2	17	10	12	28	69	71	1.9
6+0.191	5.8 E MILES CITY	CUSTER	12	54	54	2.171	24	32	PMS	212	27	680	0	14	10	11	29	64	73	.8
8+0.358	8.0 E MILES CITY	CUSTER	12	62	62	.659	32	32	PMS	175	22	365	10	25	10	1	29	75	84	.2
9+0.023		COIN 100094 149+0.238-154+0.006																		
18+0.648	DIAMOND RING INTCH	CUSTER	12	71	71	.579	26	26	PMS	168	21	671	9	28	10	0	30	77	85	.1
19+0.177	0.6 E DIAMOND INT	CUSTER	12	29	41	3.286	24	26	BST	168	21	889	0	10	5	4	30	49	61	1.7
22+0.374	PRAIRIE CO LINE	PRAIRIE	4	31	57	7.332	22	26	RMS	130	16	456	0	13	6	2	29	50	64	3.7
29+0.626	5.6 W TERRY	PRAIRIE	4	66	66	1.561	28	28	RMS	122	15	531	8	23	10	2	30	73	84	.4
31+0.365	4.0 W TERRY	PRAIRIE	4	31	57	4.030	24	30	RMS	157	20	530	0	14	6	1	29	50	63	2.0
35+0.431		PRAIRIE	4			1.002	CITY OF TERRY													
36+0.554	TERRY	PRAIRIE	4	31	57	1.294	24	30	RMS	110	15	951	0	11	5	20	30	66	80	.4
37+0.851	1.3 E TERRY	PRAIRIE	4	55	55	.844	24	32	RMS	94	14	424	6	16	9	20	30	81	91	.2
38+0.695	2.1 E TERRY	PRAIRIE	4	31	49	5.120	24	24	RMS	85	12	312	0	14	5	1	29	49	66	2.6
43+0.815	7.2 E TERRY	PRAIRIE	4	49	49	1.041	24	32	RMS	80	10	616	5	16	8	0	30	59	76	.4
44+0.856		COIN 100094 184+0.893-191+0.081																		
	ROUTE TOTAL					34.103													42	14.4
	SYSTEM TOTAL					34.103													42	14.4

SUFFICIENCY RATINGS BY SECTIONS -- STATEWIDE

RT-#	MILEPOINT	SECTION DESCRIPTION	COUNTY	FIN DIST	YEARS BT	YEARS IM	SECTION LENGTH	WIDTH SRF	WIDTH RDY	SUF TYP	ADT	DHV	SERVICE VOLUME	FND	SRF	ORN	SAF	CAP	TOT	ADJ	DEFIC MILES
P017	19+0.177	0.6 E DIAMOND INT	CUSTER	12	23	41	3.286	24	26	BST	168	21	889	0	10	5	4	30	49	61	1.7
P017	31+0.365	4.0 W TERRY	PRAIRIE	4	31	57	4.030	24	30	RMS	157	20	530	0	14	6	1	29	50	63	2.0
P017	22+0.374	PRAIRIE CO LINE	PRAIRIE	4	31	57	7.332	22	26	RMS	130	16	456	0	13	6	2	29	50	64	3.7
P017	38+0.695	2.1 E TERRY	PRAIRIE	4	31	49	5.120	24	24	RMS	85	12	312	0	14	5	1	29	49	66	2.6
P017	0+0.000	MILES CITY LIMITS	CUSTER	12	48	48	6.191	28	30	RMS	653	84	631	2	17	10	12	28	69	71	1.9
P017	6+0.191	5.8 E MILES CITY	CUSTER	12	54	54	2.171	24	32	PMS	212	27	680	0	14	10	11	29	64	73	.8
P017	43+0.815	7.2 E TERRY	PRAIRIE	4	49	49	1.041	24	32	RMS	80	10	616	5	16	8	0	30	59	76	.4
P017	36+0.554	TERRY	PRAIRIE	4	31	57	1.294	24	30	RMS	110	15	951	0	11	5	20	30	66	80	.4
P017	29+0.626	5.6 W TERRY	PRAIRIE	4	66	66	1.561	28	28	RMS	122	15	531	8	23	10	2	30	73	84	.4
P017	8+0.358	8.0 E MILES CITY	CUSTER	12	62	62	.659	32	32	PMS	175	22	365	10	25	10	1	29	75	84	.2
P017	18+0.648	DIAMOND RING INTCH	CUSTER	12	71	71	.579	26	26	PMS	168	21	671	9	28	10	0	30	77	85	.1
P017	37+0.851	1.3 E TERRY	PRAIRIE	4	55	55	.844	24	32	RMS	94	14	424	6	16	9	20	30	81	91	.2
		SYSTEM TOTAL					34.103													42	14.4

Figure 7-2. LIST-BY-SECTION and LIST-BY-RATING.

LIST-BY-RATING accesses the following files:

DD Statement	File
SUFFREP	Sufficiency report file
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages
SORTIN	Scratch file for sort
SORTOUT	Scratch file for sort
SORTWKØ1	Scratch file for sort
SORTWKØ2	Scratch file for sort
SORTWKØ3	Scratch file for sort
SORTLIB	Sort library
SYSOUT	Printed output from sort

The LIST-BY-DISTRICT Program

LIST-BY-DISTRICT prints a formatted listing of the sufficiency report file in order by financial district. Use of the program is just like LIST-BY-SECTION.

Figure 7-3 shows a sample output from LIST-BY-DISTRICT. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:LIST-BY-DISTRICT,REPORT=SUFFICIENCY,DATA=PRIMARY=17
/*
```

LIST-BY-DISTRICT accesses the same files as shown above for LIST-BY-RATING.

The RATING-BY-DISTRICT Program

RATING-BY-DISTRICT prints a summary of sufficiency ratings broken down by financial district. Use of the program is just like LIST-BY-SECTION.

Figure 7-3 shows a sample output from RATING-BY-DISTRICT. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:RATING-BY-DISTRICT,REPORT=SUFFICIENCY,DATA=PRIMARY=17
/*
```

RATING-BY-DISTRICT accesses the same files as shown above for LIST-BY-SECTION

FINANCIAL DISTRICT NUMBER 12

RT-#	MILEPOINT	SECTION DESCRIPTION	COUNTY	FIN DIST	YEARS BT	YEARS IM	SECTION LENGTH	WIDTH SRF	WIDTH RDY	SUF TYP	ADT	DHV	SERVICE VOLUME	FND	SRF	DRN	SAF	CAP	TOT	ADJ	DEFIC MILES
P017	19+0.177	0.3 E DIAMOND INT	CUSTER	12	29	41	3.286	24	26	BST	168	21	889	0	10	5	4	30	49	61	1.7
P017	0+0.000	MILES CITY LIMITS	CUSTER	12	48	48	6.191	28	30	RMS	653	84	631	2	17	10	12	28	69	71	1.9
P017	6+0.191	5.3 E MILES CITY	CUSTER	12	54	54	2.171	24	32	PMS	212	27	680	0	14	10	11	29	64	73	.8
P017	8+0.358	3.0 E MILES CITY	CUSTER	12	62	62	.659	32	32	PMS	175	22	365	10	25	10	1	29	75	84	.2
P017	18+0.643	DIAMOND RING INTCH	CUSTER	12	71	71	.579	26	26	PMS	168	21	671	9	28	10	0	30	77	85	.1
		DISTRICT TOTAL					12.886													36	4.7
		SYSTEM TOTAL					34.108													42	14.4

MILES OF RURAL HIGHWAY
DISTRIBUTED BY ADJUSTED SUFFICIENCY RATING
IF TEN PERCENT INCREMENTS

PERCENT SUFFICIENT		1	2	3	4	5	6	7	8	9	10	11	12	TOTALS	ACCUM. TOTALS
0-10 PERCENT	MILES														
	%														
11-20 PERCENT	MILES														
	%														
21-30 PERCENT	MILES														
	%														
31-40 PERCENT	MILES														
	%														
41-50 PERCENT	MILES														
	%														
51-60 PERCENT	MILES														
	%														
61-70 PERCENT	MILES				16.4								3.2	19.7	19.7
	%				77.7								25.5	58.0	58.0
71-80 PERCENT	MILES				2.3								8.3	10.6	30.4
	%				11.0								64.9	31.4	89.3
81-90 PERCENT	MILES				1.5								1.2	2.7	33.2
	%				7.4								9.6	8.2	97.5
91-100 PERCENT	MILES				.8									.8	34.1
	%				4.0									2.5	100.0

Figure 7-3. LIST-BY-DISTRICT and RATING-BY-DISTRICT.

The DEF-MILES-BY-COUNTY Program

DEF-MILES-BY-COUNTY summarizes deficient mileage by county and by financial district. Use of the program is just like LIST-BY-SECTION.

Figure 7-4 shows a partial output from a DEF-MILES-BY-COUNTY run. The job setup for this output was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:DEF-MILES-BY-COUNTY,REPORT=SUFFICIENCY,DATA=PRIMARY=17
/*
```

DEF-MILES-BY-COUNTY accesses the same files as shown above for LIST-BY-SECTION.

The MAP-TABLES Program

MAP-TABLES prints a listing by section in a format used for identifying sections in the maps that are included in the annual sufficiency report. Use of the program is just like LIST-BY-SECTION except that the PRINTER-DD parameter has no effect. Printed output is always written to the DD statement WHITE8 which specifies an all-white form printed 8 lines per inch double spaced.

Figure 7-5 shows a sample output from MAP-TABLES. The job setup for this output was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISSUFF
//SYSIN DD *
:MAP-TABLES,REPORT=SUFFICIENCY,DATA=PRIMARY=17
/*
```

MAP-TABLES accesses the following files:

DD Statement	File
SUFFREP	Sufficiency report file
WHITE8	Printed output (can be changed by OUTPUT-FILE)
SYSPRINT	IBM error messages

----- RURAL DEFICIENT MILEAGE BY COUNTY -----

FINANCIAL DISTRICT	-----COUNTY----	RURAL MILEAGE	DEFICIENT MILEAGE	PERCENT DEFICIENT
1	FLATHEAD	0.000	0.000	00.0
1	LAKE	0.000	0.000	00.0
1	LINCOLN	0.000	0.000	00.0
1	TOTAL	0.000	0.000	00.0
2	BLAINE	0.000	0.000	00.0
2	GLACIER	0.000	0.000	00.0
2	HILL	0.000	0.000	00.0
2	LIBERTY	0.000	0.000	00.0
2	TOOLE	0.000	0.000	00.0
2	TOTAL	0.000	0.000	00.0
3	DANIELS	0.000	0.000	00.0
3	PHILLIPS	0.000	0.000	00.0
3	ROOSEVELT	0.000	0.000	00.0
3	SHERIDAN	0.000	0.000	00.0
3	VALLEY	0.000	0.000	00.0
3	TOTAL	0.000	0.000	00.0
4	DAWSON	0.000	0.000	00.0
4	MCCONE	0.000	0.000	00.0
4	PRAIRIE	21.222	9.738	45.9
4	RICHLAND	0.000	0.000	00.0
4	WIBAUX	0.000	0.000	00.0
4	TOTAL	21.222	9.738	45.9
5	FERGUS	0.000	0.000	00.0
5	GARFIELD	0.000	0.000	00.0
5	PETROLEUM	0.000	0.000	00.0
5	TOTAL	0.000	0.000	00.0
6	CASCADE	0.000	0.000	00.0
6	CHOUTEAU	0.000	0.000	00.0
6	JUDITH BASIN	0.000	0.000	00.0
6	PONDERA	0.000	0.000	00.0
6	TETON	0.000	0.000	00.0
6	TOTAL	0.000	0.000	00.0

Figure 7-4. DEF-MILES-BY-COUNTY Program.

BEGINNING MILE POST	0.0	6.1	8.3	9.0	18.6	19.1	22.3	29.6	31.3	35.4	36.5	37.8	38.6	43.8	44.8
LENGTH	6.1	2.1	0.6	10.0	0.5	3.2	7.3	1.5	4.0	1.0	1.2	0.8	5.1	1.0	25.4
YEAR BUILT	43	54	62	C	71	29	31	66	31	U	31	55	31	49	C
YEAR IMPROVED	43	54	62	O	71	41	57	66	57	R	57	55	49	49	O
SURFACE WIDTH	23	24	32	I	26	24	22	28	24	B	24	24	24	24	I
ROADWAY WIDTH	30	32	32	N	26	26	26	28	30	A	30	32	24	32	N
SURFACE TYPE	RMS	PMS	PMS	C	PMS	BST	RMS	RMS	RMS	N	RMS	RMS	RMS	RMS	C
ADT	553	212	175	I	168	168	130	122	157		110	94	85	80	I
DHV	64	27	22	D	21	21	15	15	20		15	14	12	10	D
SERVICE VOLUME	531	680	365	E	671	889	455	531	530		951	424	312	616	E
FOUNDATION - 10	2	0	10	N	9	0	0	3	0		0	6	0	5	N
SURFACE - 30	17	14	25	T	28	10	13	23	14		11	16	14	16	T
DRAINAGE - 10	10	10	10		10	5	6	10	6		5	9	5	8	
SAFETY - 20	12	11	1		0	4	2	2	1		20	20	1	0	
CAPACITY - 30	28	29	29		30	30	29	30	29		30	30	29	30	
SUFFICIENCY RATING	59	64	75		77	49	50	73	50		66	81	49	59	
ADJUSTED RATING	71	73	84		85	61	64	84	63		80	91	66	76	
DEFICIENT MILEAGE	1.91	.76	.16		.13	1.67	3.65	.42	2.01		.43	.16	2.61	.42	

Figure 7-5. MAP-TABLES Program.

CHAPTER 8

THE TRAFFIC SUBSYSTEM

Introduction

The traffic file contains traffic counts for the three years prior to the current year. Each entry (or "record") in the file contains the following data elements:

- Route system and route number
- Milepoint of count
- Actual/estimated code
- Data for three years:
  - Year
  - AADT (average annual daily traffic)
  - Percent of commercial vehicles
  - Percent of pickups
  - Percent of out-of-state vehicles
- Future factor
- Design hour volume
- Remark
- Effective date

The traffic subsystem programs also utilize the roadlog and true mileage files as input.

Standard User Input - Traffic Subsystem

Because the Highway Information System has been designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the traffic programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISTRAF
//SYSIN DD *
   One or more commands
/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed by the computer.

The job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, you can direct the output to the terminal as shown in the following job setup:

```
// JOB
// EXEC HISTRAF,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *

   One or more commands

/*
```

With this job setup, the output is directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PARM='PRINTER-DD=CRJE' instead of PARM='OUTPUT-FILE=CRJE'.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Traffic Subsystem

A command facility is provided in the Highway Information System to improve the man-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section describes the most frequently used parameters that can be coded on traffic commands.

The DATA, START-MILEPOINT, and END-MILEPOINT Parameters - These parameters provide a flexible means of selecting roadways based on route system, route number, and/or milepoint. They are described in detail in chapter 1 of this manual.

The MAX-#-ENTRIES Parameter - This parameter is implemented primarily as protection to the user and as an aid in program debugging. It places a limit on the number of traffic records accepted on input (only records that meet all of the specified selection criteria are counted).

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a very flexible means for selecting bridges. The SELECT-DD parameter provides the capability of choosing traffic counts that meet selection criteria based on the contents of nearly any data element(s) in the traffic and roadlog files. The SELECT-DD software is implemented as a complete separate subsystem in the Highway Information System, and detailed information on its use is included in chapter 12.

The TRF-TYPES Parameter - TRF-TYPES provides a means of selecting traffic records by record type. Each record type is assigned a unique number:

1 Major section break records
2 Minor section break records
4 Descriptor records

Specify the sum of the desired record types. For example, to process all major and minor break records, code TRF-TYPES=3.

The LIST Program

The LIST program prints information from the traffic, roadlog, and true mileage files. One line is printed for each traffic record processed.

To use the program, prepare a command that specifies the program name LIST. Include the parameter FILE=TRAFFIC and include a DATA parameter (see chapter 1). You may if you wish include any of the following parameters (these are described in the preceding section)

START-MILEPOINT and/or END-MILEPOINT
TRF-TYPES
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

You may also include the following parameter:

LIST= $\left\{\begin{matrix} 2 \\ \underline{1} \end{matrix}\right\}$

LIST=1 (the default) prints the three years data stored in the file. LIST=2 prints only the most recent two years plus the current-year data being added.

You may also include any of the print parameters described in chapter 1 on the command.

Figure 8-1 shows a sample output from the LIST program (as well as one from the DUMP program). The output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:LIST,FILE=TRAFFIC,DATA=PRIMARY=17
/*
```

The LIST program accesses the following files:

DD Statement	File
TRAFFIC	Traffic file
ROADLOG	Roadlog file
TRUMILE	True mileage file
TABLES	Tables - if a select statement is included
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The DUMP Program

The DUMP program prints an unformatted listing of the file. One record is printed per line. The records are printed in a format very close to that of the traffic data card (see Highway Information System Release 4.∅ - Data Coding)

Use of the DUMP program is identical to the LIST program except that the parameter LIST is not used.

Figure 8-1 shows a sample output from the DUMP program (as well as one from the LIST program). The output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:DUMP,FILE=TRAFFIC,DATA=PRIMARY=17
/*
```

The DUMP program accesses the following files:

DD Statement	File
TRAFFIC	Traffic file
ROADLOG	Roadlog file - if referred to in a select statement
TABLES	Tables - if a select statement is included
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

```
HIS TRAFFIC FILE LISTING -- 1973 THROUGH 1975

                    TRUE                                         ----FIRST YEAR----    ---SECOND YEAR----   ----THIRD YEAR----
------KEY------   MILEAGE  ------SECTION DESCRIPTION--------  YR  ADT  OUT PIC COM YR  ADT  OUT PIC COM YR  ADT  OUT PIC COM FUT DHV

P00017000+0.000          JCT FAP 2 E OF MILES CITY          73 1265  49 299 149 74 1160  36 222 112 75 1087  34 320 127      130 W
P00017006+0.014    6.014                                    73  233  49 299 149 74  236  36 222 112 75  245  34 320 127      130 F
P00017009+0.022    9.020 JCT I 94                           73  157  49 299 149 74  161  36 226 112 75  167  34 320 127      130 W
P00017009+0.023          JOIN I00094 149+0.238-159+0.648                                                                          C
P00017018+0.648   18.052 JCT I 94 DIAMOND RING INT          73   36  49 299 149 74  138  36 226 112 75  168  34 320 127      130 W
P00017022+0.374   22.927 CUSTER CO LN PRAIRIE CO            73   36  49 299 149 74  138  36 226 112 75  168  34 320 127      130 W
P00017028+0.623   29.249                                    73   45  49 299 149 74   46  36 226 112 75  101  34 320 127      130 R
P00017035+0.431   35.350 ENT TERRY CL SPRING ST             73  160  49 299 149 74  285  36 226 112 75  182  34 320 127      130 T
P00017035+0.442   35.351                                                                                                          M
P00017036+0.554   35.352 LV TERRY CL                        73 1000 114 318 107 74  630 105 343  96 75  703 110 328 130      130 W
P00017036+0.557   35.355                                    73 1000 114 318 107 74  630 105 343  96 75  703 110 328 130      130 F
P00017036+0.560   35.356 JCT FAS 253 PRAIRIE COUNTY         73   80  49 318 107 74  104  36 226 112 75  116  34 320 127      130 W
P00017038+0.555   38.353                                    73   49  49 318 107 74  105  36 226 112 75   91  34 320 127      150 R
P00017044+0.855   45.150 JCT I 94 AT FALLON                 73   30  49 318 107 74   64  36 226 112 75   79  34 320 127      130 W
P00017044+0.855          JOIN I00094 184+0.803-210+0.371                                                                          C
P00017070+0.435   70.544 JCT I 94 W GLENDIVE INT            73 3319  51 329 130 74 3320  51 329 130 75 3583  51 329 130      120 W
```

```
HIS TRAFFIC FILE DUMP

P00017000+0.00073 1265 4929914974 1160 3632211275 1087 34320127                     130W
P00017006+0.01473  233 4929914974  236 3622211275  245 34320127                     130R
P00017009+0.022 73  157 4929914974  161 3622611275  167 34320127                    130W
P00017009+0.0231                                                                       C
P00017018+0.648 73   36 4929914974  138 3622611275  158 34320127                    130W
P00017022+0.374 73   36 4929914974  138 3622611275  168 34320127                    130W
P00017028+0.623 73   45 4929914974   46 3622611275  101 34320127                    130R
P00017035+0.431 73  160 4929914974  285 3622611275  182 34320127                    130T
P00017035+0.442                                                                        M
P00017036+0.554 73 100011431810774  630105343 9675  703110328130                    130W
P00017036+0.557 73 100011431810774  630105343 9675  703110328130                    130R
P00017036+0.560 73   80 4931810774  104 3622611275  116 34320127                    130W
P00017038+0.555 73   49 4931810774  105 3622611275   91 34320127                    150R
P00017044+0.855 73   30 4931810774   64 3622611275   79 34320127                    130W
P00017044+0.855                                                                        C
P00017070+0.435 73 3319 5132913074 3320 5132913075 3583 51329130                    120W
```

Figure 8-1. Traffic LIST and DUMP Programs.

The Traffic Report File

The traffic report file is a computer-generated file that must be created yearly. It contains vehicle miles and section length information on the traffic sections as well as route and system totals. The file is used for producing the annual traffic-by-sections report, sufficiency-by-sections report, accident-by-sections report, and other reports. Information on the creation of this file can be found in the publication Highway Information System Release 4.Ø - System Maintenance.

The LIST-TRAFREP Program

LIST-TRAFREP prints information from the traffic report file. To use the program, prepare a command that specifies the LIST-TRAFREP program. Include a DATA parameter and optionally a START-MILEPOINT and/or an END-MILEPOINT parameter (see chapter 1). You may also include any of the print parameters described in chapter 1.

Figure 8-2 shows a sample output from LIST-TRAFREP. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:LIST-TRAFREP,DATA=PRIMARY=17
/*
```

LIST-TRAFREP accesses the following files:

DD Statement	File
TRAFREP	Traffic report file
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The TRAFFIC-BY-SECTIONS Program

TRAFFIC-BY-SECTIONS prints the annual traffic-by-sections report using the traffic report file as input. Be sure that the report file has been created as mentioned above before utilizing this program.

To use the program, prepare a command that specifies the TRAFFIC-BY-SECTIONS program and include the parameter REPORT=TRAFFIC. Include a DATA parameter and optionally a START-MILEPOINT and/or an END-MILEPOINT parameter. You may also include any of the print parameters described in chapter 1.

HIS TRAFFIC REPORT FILE LISTING

ROUTE NUMBER	MILEPOINT		SECTION LENGTH	1973 ALL	1973 FOREIGN	1973 PICKUPS	1973 COMM	1974 ALL	1974 FOREIGN	1974 PICKUPS	1974 COMM	1975 ALL	1975 FOREIGN	1975 PICKUPS	1975 COMM
P00017	000+0.000	W	9.020	5090	249	1522	758	4794	172	1414	536	4624	157	1479	587
P00017	009+0.022	D													
P00017	009+0.023	C													
P00017	018+0.648	W	3.365	139	6	41	20	533	19	120	59	649	22	207	82
P00017	022+0.374	W	12.923	932	45	278	138	1674	60	378	137	1784	60	570	226
P00017	035+0.431	T	1.002												
P00017	036+0.554	W	.006	6		1		3		1		4		1	
P00017	036+0.550	W	8.292	377	18	120	40	740	26	167	82	741	25	237	94
P00017	044+0.355	D													
P00017	044+0.356	C													
P00017	070+0.435	E													
P00017	999RURAL	E	34.105	6546	321	1964	959	7746	279	2081	867	7804	265	2497	991

FAP 17

MILE POST	SECTION DESCRIPTION	COUNTY	TRAFFIC TYPE	SECTION LENGTH	ADT 73	ADT 74	ADT 75	VEH MILES
0+0.000	JCT FAP 2 E OF MILES CITY	CUSTER	ALL VEHICLES	9.020	564	531	512	4624
			COMMERCIAL		84	59	65	587
			OUT OF STATE		27	19	17	157
9+0.022	JCT I 94							
	COIN I00094 149+0.238-154+0.006							
18+0.643	JCT I 94 DIAMOND RING INT	CUSTER	ALL VEHICLES	3.865	35	138	168	649
			COMMERCIAL		5	15	21	82
			OUT OF STATE		1	4	5	22
22+0.374	CUSTER CO LN PRAIRIE CO	PRAIRIE	ALL VEHICLES	12.923	72	129	138	1784
			COMMERCIAL		10	14	17	226
			OUT OF STATE		3	4	4	60
35+0.431	ENT TERRY CL SPRING ST	PRAIRIE		1.002				

Figure 8-2. LIST-TRAFREP and TRAFFIC-BY-SECTIONS.

Figure 8-2 shows a sample output from TRAFFIC-BY-SECTIONS. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:TRAFFIC-BY-SECTIONS,REPORT=TRAFFIC,DATA=PRIMARY=17,
:  END-MILEPOINT=Ø36+Ø.ØØØ
/*
```

TRAFFIC-BY-SECTIONS accesses the following files:

DD Statement	File
TRAFREP	Traffic report file
ROADLOG	Roadlog file
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The SUMMARY-BY-ROUTES Program

SUMMARY-BY-ROUTES prints a breakdown by route with system totals of traffic data taken from the traffic report file. To use the program, prepare a command that specifies the SUMMARY-BY-ROUTES program and include the parameter REPORT=TRAFFIC. Include a DATA parameter in the following format:

$$\text{DATA=} \left\{ \begin{array}{l} \text{INTERSTATE} \\ \text{PRIMARY} \\ \text{SECONDARY} \\ \text{URBAN} \end{array} \right\}$$

You may also include any of the print parameters described in chapter 1.

Figure 8-3 shows a sample output from SUMMARY-BY-ROUTES. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:SUMMARY-BY-ROUTES,REPORT=TRAFFIC,DATA=INTERSTATE
/*
```

SUMMARY-BY-ROUTES accesses the same files shown above for LIST-TRAFREP.

1975
INTERSTATE RURAL ROUTE SUMMARY

ROUTE NO	RURAL MILEAGE	AVERAGE DAILY TRAFFIC: ALL VEHICLES	COMMER-CIAL	OUT OF STATE	*****VEHICLE MILES*****: ALL VEHICLES	COMMER-CIAL	OUT OF STATE	% OF TOTAL VEH. MILES: ALL VEHICLES	COMMER-CIAL	OUT OF STATE
00015	391.2	1888	306	379	738597	119647	148098	23.15	19.47	17.66
00090	527.3	3557	709	1053	1928328	373995	555108	60.45	60.85	66.19
00094	245.8	2127	492	551	523002	120392	135488	16.40	19.69	16.15
NET TOTAL	1164.4	2740	528	720	3189927	614033	838694	100.00	100.00	100.00

RURAL MILEAGE AND VEHICLE MILES OF TRAVEL BY COUNTY AND SYSTEM

COUNTY	FEDERAL AID SYSTEM: INTERSTATE MILES	INTERSTATE VEHICLE MILES	PRIMARY MILES	PRIMARY VEHICLE MILES	SECONDARY MILES	SECONDARY VEHICLE MILES	TOTAL RURAL MILES	TOTAL RURAL VEHICLE MILES
1	85.350	101760.040	91.916	33702.635	131.662	17481.161	308.928	152943.836
2	79.475	195659.793	51.318	50531.961	174.444	43258.852	315.737	289450.606
3			53.911	85309.947	201.999	34106.624	255.910	119416.571
4	5.704	18152.925	79.615	94575.076	51.085	8362.407	136.404	121090.408
5			103.599	138417.758	74.900	30773.044	178.499	169190.802
6			49.994	27222.535	153.337	12424.191	203.331	39646.726
7	59.041	178522.555	167.407	197616.743	127.068	33534.636	353.516	409673.944
8			72.360	89935.221	237.712	39067.245	310.072	129052.466
9	42.304	80983.338	131.143	74191.708	75.083	8996.033	249.030	164171.079
10			46.970	15679.333	66.693	11119.308	113.663	26798.641
11	43.199	88612.074	85.564	71916.934	106.790	15559.511	235.553	176088.519
12	14.213	56626.935	45.546	73665.629	38.257	10858.552	98.021	141151.116
13			83.113	42993.393	101.680	6960.327	184.793	49958.720
14			179.033	121940.712	145.958	29010.028	325.036	150950.740
15			208.221	401166.544	160.643	103096.487	368.864	504263.031
16	42.279	196958.974	135.878	131372.017	163.210	76988.495	341.367	455319.486
17			135.362	37246.158	127.566	7677.725	262.928	44923.883
18			147.870	150763.535	153.447	32149.225	301.317	182912.760
19			39.661	33012.346	16.428	1354.934	56.089	34367.280
20	27.755	101779.416	80.265	25038.210	32.254	2783.659	140.274	129601.285
21			78.213	114260.041	186.702	27258.388	264.915	141518.429
22	94.311	207330.857	28.399	13848.609	72.582	27396.059	195.792	248575.525

Figure 8-3. Traffic SUMMARY-BY-ROUTES and SUM-BY-COUNTY.

The SUM-BY-COUNTY Program

SUM-BY-COUNTY prints a summary of traffic information broken down by county. To use the program, prepare a command that specifies the SUM-BY-COUNTY program and include the parameter REPORT=TRAFFIC. You may include any of the print parameters described in chapter 1.

Figure 8-3 shows a sample output from SUM-BY-COUNTY. The job setup for this output was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:SUM-BY-COUNTY,REPORT=TRAFFIC
/*
```

SUM-BY-COUNTY accesses the following files:

DD Statement	File
TRAFREP	Traffic report file
ROADLOG	Roadlog file
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

CHAPTER 9

THE TRUE MILEAGE SUBSYSTEM

Introduction

The true mileage file contains the actual locations of reference posts. Each entry ("record") in the file contains the location of one reference post. Each record contains the following data elements:

- Route system and route number
- Reference post number
- Location of reference post (distance in miles from beginning of route)
- Effective date
- Most recent date of update

The true mileage subsystem is implemented within the traffic subsystem. Refer to the sections "Standard User Input - Traffic Subsystem" and "Commands - Traffic Subsystem" in chapter 8 for an introduction to the use of the subsystem.

The LIST Program

The LIST program prints true mileage records, one to a line. In addition to the above data elements, the program calculates and prints the distance between reference posts.

To use the LIST program, prepare a command that specifies the program name LIST and that includes the parameter FILE=TRUMILE. Include a DATA parameter (see chapter 1) to identify which route(s) are to be printed. You may include the parameters START-MILEPOINT and END-MILEPOINT as needed (code only the reference post portion as in END-MILEPOINT=Ø6Ø). You may also include any of the print parameters described in chapter 1 as needed.

Figure 9-1 shows a sample output from LIST. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISTRAF
//SYSIN DD *
:LIST,FILE=TRUMILE,DATA=PRIMARY=17,END-MILEPOINT=Ø4Ø
/*
```

HIS TRUE MILEAGE LISTING

ROUTE NUMBER	REFERENCE POST	TRUE MILEAGE	DIST FROM PRECEDING	EFFECTIVE DATE	DATE OF UPDATE
P00017	0	0.000			
P00017	1	0.995	0.995		11/19/75
P00017	2	2.006	1.011		11/19/75
P00017	3	2.993	0.987		11/19/75
P00017	4	4.007	1.014		11/19/75
P00017	5	4.999	0.992		11/19/75
P00017	6	6.000	1.001		11/19/75
P00017	7	7.008	1.008		11/19/75
P00017	8	8.004	0.996		11/19/75
P00017	9	8.998	0.994		11/19/75
P00017	10	9.984	0.986		11/19/75
P00017	11	10.987	1.003		11/19/75
P00017	12	11.987	1.000		11/19/75
P00017	13	12.991	1.004		11/19/75
P00017	14	13.953	0.962		11/19/75
P00017	15	14.414	0.461		11/19/75
P00017	16	15.414	1.000		11/19/75
P00017	17	16.414	1.000		11/19/75
P00017	18	18.414	2.000		11/19/75
P00017	19	19.464	1.050		11/19/75
P00017	20	20.497	1.033		11/19/75
P00017	21	21.530	1.033		11/19/75
P00017	22	22.553	1.023		11/19/75
P00017	23	23.565	1.012		11/19/75
P00017	24	24.571	1.006		11/19/75
P00017	25	25.587	1.016		11/19/75
P00017	26	26.594	1.007		11/19/75
P00017	27	27.610	1.016		11/19/75
P00017	28	28.626	1.016		11/19/75
P00017	29	29.633	1.007		11/19/75
P00017	30	30.455	0.822		11/19/75
P00017	31	31.455	1.000		11/19/75
P00017	32	32.455	1.000		11/19/75
P00017	33	33.414	0.959		11/19/75
P00017	34	34.383	0.969		11/19/75
P00017	35	35.419	1.036		11/19/75
P00017	36	36.298	0.879		11/19/75
P00017	37	37.295	0.997		11/19/75
P00017	38	38.295	1.000		11/19/75
P00017	39	39.295	1.000		11/19/75
P00017	40	40.295	1.000		11/19/75

Figure 9-1. True Mileage LIST Program.

The true mileage LIST program accesses the following files:

DD Statement	File
TRUMILE	True mileage file
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	Used for IBM error messages

CHAPTER 1Ø

THE URBAN SIGN INVENTORY SUBSYSTEM

Introduction

The urban sign inventory subsystem is designed for use by any cities that desire an inventory of their signs. Each city that uses the subsystem has its own separate file.

Data is stored on two separate levels: the "assembly" level and the "sign" level. An assembly is a set of one or more posts supporting one or more signs. Information stored at the assembly level is applicable to all of the signs on an assembly. Information stored at the sign level is applicable to a single sign.

Information stored at the assembly level (one entry for each assembly) includes:

- Assembly number and tie-breaker (a unique identifier for each assembly)
- Type of most recent maintenance
- Date and time of most recent maintenance
- Date of original installation or inventory
- Post condition
- Post position
- Assembly location
- X- and y-coordinates
- Map number
- Assembly type
- Post type
- Number of posts
- Visibility
- X- and y-feet
- Change in mounting height or lateral clearance
- Cost fields
- Date of update

Information stored at the sign level (one entry for each sign) includes:

- Assembly number and tie-breaker
- Sign number (a unique identifier for each sign on an assembly)
- Date and time of most recent maintenance
- Date of original installation or inventory
- Sign condition
- Sign position
- Uniform/non-uniform code
- Direction of facing
- Visibility on other routes
- Side of street
- Functional classification
- Federal aid system
- Maintenance responsibility

Sign code number and supplemental code
Sign color
Letter type
Material
Face
Shape
Horizontal dimension
Vertical dimension
Mounting height
Lateral clearance
Sign age
Cost fields
Date of update

In addition, verbal remarks can be stored at either the assembly level or at the sign level.

The following additional files, described in chapter 11, are accessed by urban sign programs:

The city route cross-reference file - contains an inventory of all city streets including the assigned street numbers.

The sign code cross-reference file - contains a list of all the valid sign types.

The grid table - contains an inventory of all the intersections in the city.

<u>Standard User Input - Urban Sign Subsystem</u>

Because the Highway Information System has been designed with the user in mind, user input has been standardized to as great a degree as possible. To run most of the urban sign programs, the following job setup is submitted to the computer:

```
// JOB
// EXEC HISUSN
//SYSIN DD *

   One or more commands

/*
```

A JOB card must be assigned to you by the Data Processing Bureau - it contains accounting information needed by the computer.

The job setup can be either punched onto cards (begin all cards in column 1) and submitted at the Data Processing Bureau, or can be submitted via a CRJE terminal. The job setup can also be submitted on diskettes but additional control information may be necessary. When submitting a run from a CRJE terminal, you can direct the output to the terminal. A sample job setup is:

```
// JOB
// EXEC HISUSN,PARM='OUTPUT-FILE=CRJE'
//SYSIN DD *

   One or more commands

/*
```

PARM='OUTPUT-FILE=CRJE' requests that the output be directed to both the computer's printer and to your terminal. To receive the output at your terminal only, specify PARM='PRINTER-DD=CRJE'.

A number of print formatting options - page numbering, obtaining more than one copy, special print forms, etc. - are available. These options are described in chapter 1 of this manual.

Commands - Urban Sign Subsystem

A command facility is provided in the Highway Information System to improve the user-machine interface. The commands allow users to request computer runs by means of meaningful symbols in standardized formats. Chapter 1 contains a description of the standard command format. The remainder of this section describes the parameters that can be used on urban sign commands.

The START-ASSEMBLY and END-ASSEMBLY Parameters - These parameters can be used to select assemblies or signs based on assembly number. START-ASSEMBLY limits access to those assemblies and signs that have an assembly number equal to or larger than the specified value. END-ASSEMBLY limits access to those assemblies and signs that have an assembly number equal to or smaller than the specified value. Code a 6-character value that includes the tie-breaker as the first character and the assembly number in the remaining 5 characters. An example is START-ASSEMBLY=EØØ15Ø. If your city does not utilize tie-breaker codes, the option must be enclosed in quotes and the first character left blank, as in START-ASSEMBLY=' ØØ15Ø'.

The DATE Parameter - DATE allows you to produce summaries based on historical data rather than on current data. Code a date in mm/dd/yy format (including leading zeroes as needed) as in DATE=Ø1/Ø1/75. The output contains results based on the inventory's status as of the specified date. In the absence of this parameter, the results are based on the current data stored in the files.

The ACCESS Parameter - The ACCESS parameter has the following format:

$$\text{ACCESS=}\left\{\begin{array}{l}\text{HISTORICAL}\\ \underline{\text{CURRENT}}\end{array}\right\}$$

ACCESS=CURRENT, the default, prints only the most recent data in a listing. ACCESS=HISTORICAL prints all historical data as well as the current data.

The CITY Parameter - The CITY parameter is required on all urban sign commands. It identifies the city whose file is being accessed. A list of city names is provided in table 2-1 in chapter 2.

The MAX-#-ENTRIES Parameter - MAX-#-ENTRIES places a limit on the number of assemblies that can be processed. Only assemblies that meet all of the specified selection criteria are counted.

The SELECT-DD and SELECT-SIZE Parameters - These parameters provide a flexible means of selecting assemblies and/or signs for inclusion in listings or summaries. Detailed information on the use of the select facilities is included in chapter 12.

The LIST Program

The LIST program prints a listing of information from the urban sign files. It can be instructed to print (1) assembly information only with or without verbal remarks, (2) sign information only with or without verbal remarks, or (3) assembly information and sign information with or without verbal remarks and with or without cost fields.

To use the LIST program, prepare a command that specifies the LIST program and that includes the parameter FILE=URBAN-SIGN. You may include the following parameter:

$$\text{LIST=}\left\{\begin{array}{l}\text{ASSEMBLY}\\ \text{SIGN}\\ \text{COST}\\ \underline{\text{BOTH}}\end{array}\right\}$$

LIST=BOTH, the default, requests that both assembly and sign information be printed and that the cost fields be excluded. LIST=COST requests both assembly and sign information be printed including the cost fields. LIST=SIGN requests only sign information (including cost fields). LIST=ASSEMBLY requests only assembly information (including cost fields).

You may also include the following parameter:

$$\text{FILE-FORMAT=}\left\{\begin{array}{l}\text{NO-REMARK}\\ \underline{\text{REMARK}}\end{array}\right\}$$

FILE-FORMAT=REMARK, the default, requests that the verbal remarks be printed. FILE-FORMAT=NO-REMARK requests that the verbal remarks be omitted.

Another parameter that can be included is:

$$\text{CODE-LIST=}\left\{\begin{array}{l}\text{NO}\\ \underline{\text{YES}}\end{array}\right\}$$

CODE-LIST=YES requests that a list of codes used in the listing be printed. This code listing requires four pages of output. CODE-LIST=NO requests that the code listing be suppressed (this option is especially useful when obtaining output over a CRJE terminal).

The following parameters may be included on the command (these are described in the previous section):

START-ASSEMBLY and/or END-ASSEMBLY
DATE
ACCESS
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

A CITY parameter must be included on the command (see table 2-1 in chapter 2). You may also include any of the print parameters described in chapter 1.

Figure 1Ø-1 shows one page of a LIST run. The job setup for this output was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:LIST,FILE=URBAN-SIGN,CITY=GREAT-FALLS,LIST=BOTH,
:      FILE-FORMAT=REMARK,CODE-LIST=NO,
:      START-ASSEMBLY=UØØØØ7,END-ASSEMBLY=UØØØ13
/*
```

The LIST program accesses the following files:

DD Statement	File
USNxxx	Urban sign file (xxx is city number)
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The LIST-SIGNS-BY-STREET Program

LIST-SIGNS-BY-STREET prints a listing of signs in the order they are seen when travelling along a street. Two listings are produced for two-way streets - one for each direction of travel. One listing is produced for one-way streets.

The program can be instructed to produce listings for all of the streets in a city or to produce listings only for selected streets. It can also be instructed to limit its listing to a portion of the city rather than to the entire city.

To use the program, prepare a command that specifies the LIST-SIGNS-BY-STREET program. Include a CITY parameter (see table 2-1 in chapter 2). You may include any of the following selection parameters (these are described earlier in this chapter in the section "Commands - Urban Sign Subsystem"):

START-ASSEMBLY and/or END-ASSEMBLY
DATE
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

To list all of the streets you must include the parameter TYPE-RUN=ALL-STREETS. To list selected streets you must include the parameter TYPE-RUN=SELECTED-STREETS and you must include a DDNAME parameter (this parameter is described below). You

```
URBAN SIGN FILE LISTING - APR 13,1976   -----   CITY OF GREAT-FALLS
                                                                                P V              X-FT  Y-FT
                                                                                O !              ----------
         (     COST      )             I       LOCATION            X     Y     S S  F     F                H  L
       S -----------------                N                  MAP ASM  POST COORD COORD T I U A   E E P  C     S  E  A
       I                   H          S C  --------------   NUM TYPE TYPE ----------- S B N C S J D E  O L M F H  I  T
ASSEMB G     UPDT   DATE   O    DATE   C O P                -------------             --- E I I N A S  L E A A A  G  E
       N                   U           O N O   VISIBILITY               DIMENSION       R N D C I P  O T T C P  H  R
NUMBER # SEQ CODE MAINTAIN R  INS/INV  D O S  OTHER ROUTES     SIGN-CODE   HORZ VERT  AGE M G E T O N  R R L E E  T  L   UPDATE
---------------------------------------------------------------------------------------------------------------------------
U00007   699 AINV  1/01/75 UN  1/01/75 V 0 0               160        02                  1 0
U00007 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               201-01-00     030  030  000      0 U  01 2 3 3 1 06  03
---------------------------------------------------------------------------------------------------------------------------
U00008   699 AINV  1/01/75 UN  1/01/75 V 0 0               160        02                  1 0
U00008 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               201-01-00     030  030  000      0 U  01 2 3 3 1 07  02
---------------------------------------------------------------------------------------------------------------------------
U00009   699 AINV  1/01/75 UN  1/01/75 V 0 0               160        02                  1 0
U00009 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               202-01-41     024  030  000      0 U  02 2 3 3 3 06  10
U00009 A 999 RMRK  1/01/75 UN              SIGN REMARK --- S L 45
---------------------------------------------------------------------------------------------------------------------------
U00010   699 AINV  1/01/75 UN  1/01/75 V 0 0               160        02                  1 0
U00010 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               201-01-00     030  030  000      0 U  01 2 3 3 1 06  02
---------------------------------------------------------------------------------------------------------------------------
U00011   699 AINV  1/01/75 UN  1/01/75 V 0 0               160        02                  1 0
U00011 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               202-01-31     024  030  000      0 U  02 2 3 3 3 07  10
U00011 A 999 RMRK  1/01/75 UN              SIGN REMARK --- S L 40
---------------------------------------------------------------------------------------------------------------------------
U00012   699 AINV  1/01/75 UN  1/01/75 V 0 0               161        26                  1 0
U00012 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               201-01-00     030  030  000      0 U  01 2 3 3 1 06  05
---------------------------------------------------------------------------------------------------------------------------
U00013   699 AINV  1/01/75 UN  1/01/75 V 0 0               161        02                  1 0
U00013 A 699 SINV  1/01/75 UN  1/01/75 V 0 0               202-01-31     024  030  000      0 U  02 2 3 3 3 07  10
U00013 A 999 RMRK  1/01/75 UN              SIGN REMARK --- S L 40
```

Figure 1Ø-1. Urban Sign LIST Program.

must also include a DDNAME parameter if you wish to limit the listing to a portion of the city.

The DDNAME parameter is used to enter options regarding city areas and regarding selected streets. Code a 4- to 8-character name that begins with the characters LSBS. The name is used as a linkage to data cards included with the run that identify (1) the city area(s), and (2) the selected streets.

To limit the listing to a portion of the city, include one or more cards in the following format:

Columns	Length	Contents
1-4	4	'AREA'
5	1	blank
6-9	4	City route number
1∅	1	blank
11-14	4	City route number of intersecting street
15	1	blank
16-19	4	Length in feet of one side of a square
2∅-8∅	61	blank

The AREA card describes a square that has an intersection at its center. The two route numbers provide the location of an intersection. The length in feet provides the size of the square. Each AREA card instructs the program to limit access to those assemblies and signs that lie within the square.

To specify selected streets, include one or more cards in the following format:

Columns	Length	Contents
1-4	4	Route number
5	1	Direction
6-8∅	75	blank

Code the route number of the street you are selecting. In the direction code, code an 'A' to list signs seen in the A direction of the route, a 'B' to list signs seen in the B direction of the route, or a blank to list signs seen in both directions (two separate listings).

The following examples illustrate various types of job setups for the LIST-SIGNS-BY-STREET program:

Example 1: List all streets in a selected area.

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:LIST-SIGNS-BY-STREET,CITY=GREAT-FALLS,TYPE-RUN=ALL-STREETS,
:   DDNAME=LSBS
/*
//LSBS DD *
AREA 412Ø 12Ø8 16ØØ
/*
```

Example 2: List several selected streets.

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:LIST-SIGNS-BY-STREET,CITY=GREAT-FALLS,TYPE-RUN=SELECTED-STREETS,
:   DDNAME=LSBS1
/*
//LSBS1 DD *
421Ø
17Ø8A
17Ø8B
1975B
1142
/*
```

Example 3: List several selected streets in different areas.

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:LIST-SIGNS-BY-STREET,CITY=GREAT-FALLS,TYPE-RUN=SELECTED-STREETS,
:   DDNAME=LSBSQRXZ
/*
//LSBSQRXZ DD *
12ØØ   (entire city)
AREA 1ØØØ 2ØØØ 15ØØ
12ØØ   (only within selected area)
1422A  (only within selected area)
AREA 42ØØ 32Ø1 15ØØ
1433B  (only within selected area)
/*
```

Figure 1Ø-2 shows a sample output from the LIST-SIGNS-BY-STREET program. This output is one page from the output obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:LIST-SIGNS-BY-STREET,CITY=GREAT-FALLS,TYPE-RUN=ALL-STREETS
/*
```

The LIST-SIGNS-BY-STREET program accesses the following files:

DD Statement	File
USNxxx	Urban sign file (xxx is city number)
CTYXxxx	City route cross-reference file (xxx is city number)
GRIDTBL	Grid table
SIGNEDT	Sign code cross-reference file
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages
ddname	Data cards - if DDNAME included
SORTIN	Scratch file for sort
SORTOUT	Scratch file for sort
SORTWKØ1	Scratch file for sort
SORTWKØ2	Scratch file for sort
SORTWKØ3	Scratch file for sort
SORTLIB	Sort library
SYSOUT	Sort printed output

The SUMMARY-BY-CONDITION Program

SUMMARY-BY-CONDITION prints a set of one or more summaries based on post or sign condition. It can print the following summaries of assembly information:

Summary Number	Summary
A-Ø1	Assemblies by post position and post condition
A-Ø2	Assemblies by assembly type and post condition
A-Ø3	Assemblies by post type and post condition
A-Ø4	Assemblies by number of posts and post condition
A-Ø5	Assemblies by visibility and post condition

It can also print the following summaries of sign information:

```
    LIST SIGNS BY STREET        EAST ON E STREET                  ALL STREETS                    APR 7,1976

                                                                       F      F       L
          S                                                 V          A    F E M   C E M   S
          I                                                 I  C       C  U U D A   O T A F H
          G    SIDE                                         S  O  P    I  N N A I   L T T A A
          N     OF    ASSEMB   MAP                    POSTS I  N  O    N  I C I N   O E E C P  MOUNTING  DIMEN     VISIBILITY   DATE OF
LOCATION #   STREET  NUMBER   NUM   ASSEMBLY TYPE    # TYPE B  D  S    G  F T D T   R R R E E  HITE LAT  HOR VRT  OTHER ROUTES  INS/INV   AGE
--------------------------------------------------------------------------------------------------------------------------------------------
0905B0101-0100        U00001  001   REGULATORY        1  16 1  0  0          100 FEET BEFORE 1ST AVENUE                         01/01/75V

          A  RIGHT   R09-04-01      NO HITCH HIKING            0  0    B    1 L S  02  2 4 3 3   07  20  018 024                01/01/75V UNK
--------------------------------------------------------------------------------------------------------------------------------------------
0905BC101+0000        U00024  001   REGULATORY        0  20 0  0  0            0 FEET AFTER  1ST AVENUE                         01/01/75V

          A  OVER    R10-09-01      RIGHT ON RED AFTER STO     0  0    B    1 L S  02  3 6 3 3   18  00  024 030                01/01/75V UNK
          B  OVER    R03-04-01      DIAGRAM - NO U TURN        0  0    B    1 L S  01  2 4 1 2   18  00  024 024                01/01/75V UNK
          C  OVER    R03-04-11      NO U TURN                  0  0    B  * 1 L S  02  2 4 3 3   19  00  024 018                01/01/75V UNK
--------------------------------------------------------------------------------------------------------------------------------------------
0102A0904+0180        U00022  001   WARNING           1  02 0  0  0                         NEAR 2ND AVENUE                     01/01/75V

          B  LEFT    W11-07-01      RANGE CATTLE               8  1    N  * 3 L N  00  1 1 1 4   06  16  030 030  0905B         01/01/75V UNK
--------------------------------------------------------------------------------------------------------------------------------------------
0905B0103+0010        U00027  001   INFORMATION       1  10 2  0  0           10 FEET AFTER  3RD AVENUE                         01/01/75V

          A  RIGHT   N04-01-01      WATER 400 FT.              0  0    B    1 L S  03  1 1 1 3   04  10  024 012  0103F         01/01/75V UNK
--------------------------------------------------------------------------------------------------------------------------------------------
0905B0105+0160        U00028  001   REGULATORY        1  07 0  0  0          160 FEET AFTER  5TH AVENUE                         01/01/75V

          A  RIGHT   R02-01-92      SPEED LIMIT 70             0  0    B    1 L S  02  2 3 3 3   07  15  048 060                01/01/75V UNK
          B  RIGHT   R11-04-01      ROAD CLOSED TO THRU TR     0  0    B    1 L S  02  2 3 3 3   13  15  060 030                01/01/75V UNK
```

Figure 1∅-2. LIST-SIGNS-BY-STREET Program.

Summary Number	Summary
S-Ø1	Signs by sign position and sign condition
S-Ø2	Signs by functional class and sign condition
S-Ø3	Signs by federal aid system and sign condition
S-Ø4	Signs by maint. responsibility and sign condition
S-Ø5	Signs by color and sign condition
S-Ø6	Signs by letter type and sign condition
S-Ø7	Signs by material and sign condition
S-Ø8	Signs by face and sign condition
S-Ø9	Signs by shape and sign condition

To use the program, prepare a command that specifies the SUMMARY-BY-CONDITION program. Include a CITY parameter (see table 2-1 in chapter 2). You may include the following parameter:

$$\text{OPTION-LIST=} \begin{Bmatrix} \text{NO} \\ \underline{\text{YES}} \end{Bmatrix}$$

OPTION-LIST=YES, the default, prints a list of the options specified on the command. OPTION-LIST=NO suppresses this listing.

You may include any of the following selection parameters on the command (these are described earlier in this chapter in the section "Commands - Urban Sign Subsystem"):

START-ASSEMBLY and/or END-ASSEMBLY
DATE
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

You may also include any of the print parameters described in chapter 1.

Unless you specify otherwise, all 14 of the summaries are printed. You can specify any combination of summaries by using either the SUMMARIES or the DDNAME parameter. Use the SUMMARIES parameter to specify up to 5 summaries as in the following example:

SUMMARIES='A-Ø3,S-Ø2,S-Ø7,A-Ø1'

To specify more than 5 summaries you must use the DDNAME parameter and enter one or more data cards that list the summaries. Code in DDNAME a 4- to 8-character name that begins with SCND. List the summaries you wish on one or more data cards (begin each card in column 1) as shown in the following sample job setup:

CITY OF ALBERTON APR 13,1976

	SATIS-FACTORY	INADE-QUATE REPLACE	DAMAGED VANDAL REPLACE	DAMAGED ACIDENT REPLACE	DAMAGED NATURAL REPLACE	TOTAL REPLACE	DAMAGED VANDAL REPAIR	DAMAGED ACIDENT REPAIR	DAMAGED NATURAL REPAIR	TOTAL REPAIR	DOES NOT EXIST	TOTAL
SUMMARY A-01 - ASSEMBLIES BY POST POSITION:												
0 - SATISFACTORY	38	3				3					1	42
1 - SHOULD BE RELOCATED	1				1	1			1	1		3
TOTAL	39	3			1	4			1	1	1	45
0 - SATISFACTORY	84.44%	6.67%	0.00%	0.00%	0.00%	6.67%	0.00%	0.00%	0.00%	0.00%	2.22%	93.33%
1 - SHOULD BE RELOCATED	2.22%	0.00%	0.00%	0.00%	2.22%	2.22%	0.00%	0.00%	2.22%	2.22%	0.00%	6.67%
TOTAL	86.67%	6.67%	0.00%	0.00%	2.22%	8.89%	0.00%	0.00%	2.22%	2.22%	2.22%	100.00%
SUMMARY A-02 - ASSEMBLIES BY ASSEMBLY TYPE:												
ONE SIGN ONLY	20	3				3						23
A - ADVANCE TURN	1											1
C - CONFIRMING ROUTE	4											4
D - DIRECTION	1											1
G - GUIDE	1											1
I - INFORMATION	1											1
J - JUNCTION	1											1
R - REGULATORY	5								1	1	1	7
S - SCHOOL					1	1						1
T - TURN	1											1
W - WARNING	3											3
X - COMBINATION	1											1
TOTAL	39	3			1	4			1	1	1	45
ONE SIGN ONLY	44.44%	6.67%	0.00%	0.00%	0.00%	6.67%	0.00%	0.00%	0.00%	0.00%	0.00%	51.11%
A - ADVANCE TURN	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
C - CONFIRMING ROUTE	8.89%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	8.89%
D - DIRECTION	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
G - GUIDE	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
I - INFORMATION	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
J - JUNCTION	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
R - REGULATORY	11.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%	2.22%	2.22%	15.56%
S - SCHOOL	0.00%	0.00%	0.00%	0.00%	2.22%	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
T - TURN	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
W - WARNING	6.67%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	6.67%
X - COMBINATION	2.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%
TOTAL	86.67%	6.67%	0.00%	0.00%	2.22%	8.89%	0.00%	0.00%	2.22%	2.22%	2.22%	100.00%

Figure 1Ø-3. SUMMARY-BY-CONDITION Program.

CITY OF ALBERTON APR 13,1976

SUMMARY S-01 - SIGNS BY SIGN POSITION:

	SATIS-FACTORY	NON-STNDARD REPLACE	DAMAGED VANDAL REPLACE	DAMAGED ACIDENT REPLACE	DAMAGED NATURAL REPLACE	TOTAL REPLACE	DAMAGED VANDAL REPAIR	DAMAGED ACIDENT REPAIR	DAMAGED NATURAL REPAIR	TOTAL REPAIR	IMPROPER SHOULD REMOVE	DOES NOT EXIST	TOTAL
0 - SATISFACTORY	74	2				2			1	1	2	2	81
1 - UNSATISFACTORY	4	1				1					2		7
TOTAL	78	3				3			1	1	4	2	88
0 - SATISFACTORY	84.09%	2.27%	0.00%	0.00%	0.00%	2.27%	0.00%	0.00%	1.14%	1.14%	2.27%	2.27%	92.05%
1 - UNSATISFACTORY	4.55%	1.14%	0.00%	0.00%	0.00%	1.14%	0.00%	0.00%	0.00%	0.00%	2.27%	0.00%	7.95%
TOTAL	88.64%	3.41%	0.00%	0.00%	0.00%	3.41%	0.00%	0.00%	1.14%	1.14%	4.55%	2.27%	100.00%

SUMMARY S-02 - SIGNS BY FUNCTIONAL CLASSIFICATION:

	SATIS-FACTORY	NON-STNDARD REPLACE	DAMAGED VANDAL REPLACE	DAMAGED ACIDENT REPLACE	DAMAGED NATURAL REPLACE	TOTAL REPLACE	DAMAGED VANDAL REPAIR	DAMAGED ACIDENT REPAIR	DAMAGED NATURAL REPAIR	TOTAL REPAIR	IMPROPER SHOULD REMOVE	DOES NOT EXIST	TOTAL
0 - INTERSTATE	27	2				2							29
1 - URB PRINC ART	16	1				1					1		18
2 - URB MINOR ART	1												1
3 - URB COLLECTOR	13										1		14
4 - URBAN LOCAL	7								1	1			8
5 - RUR PRINC ART	1												1
6 - RUR MINOR ART	3										1		4
7 - RUR MAJ COLL	1												1
8 - RUR MIN COLL	2												2
9 - RURAL LOCAL	7										1	2	10
TOTAL	78	3				3			1	1	4	2	88
0 - INTERSTATE	30.68%	2.27%	0.00%	0.00%	0.00%	2.27%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	32.95%
1 - URB PRINC ART	18.18%	1.14%	0.00%	0.00%	0.00%	1.14%	0.00%	0.00%	0.00%	0.00%	1.14%	0.00%	20.45%
2 - URB MINOR ART	1.14%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%
3 - URB COLLECTOR	14.77%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%	0.00%	15.91%
4 - URBAN LOCAL	7.95%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%	1.14%	0.00%	0.00%	9.09%
5 - RUR PRINC ART	1.14%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%
6 - RUR MINOR ART	3.41%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%	0.00%	4.55%
7 - RUR MAJ COLL	1.14%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%
8 - RUR MIN COLL	2.27%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.27%
9 - RURAL LOCAL	7.95%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.14%	2.27%	11.36%
TOTAL	88.64%	3.41%	0.00%	0.00%	0.00%	3.41%	0.00%	0.00%	1.14%	1.14%	4.55%	2.27%	100.00%

Figure 1∅-4. SUMMARY-BY-CONDITION Program.

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:SUMMARY-BY-CONDITION,CITY=GREAT-FALLS,DDNAME=SCNDX
/*
//SCNDX DD *
A-Ø4,S-Ø2,S-Ø9
S-Ø1,A-Ø1
/*
```

Figure 1Ø-3 shows a sample output of A-Ø1 and A-Ø2. Figure 1Ø-4 shows a sample output of S-Ø1 and S-Ø2. The outputs were obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:SUMMARY-BY-CONDITION,CITY=ALBERTON,
:   SUMMARIES='A-Ø1,A-Ø2,S-Ø1,S-Ø2'
/*
```

SUMMARY-BY-CONDITION accesses the following files:

DD Statement	File
USNxxx	Urban sign file (xxx is city number)
ddname	Data cards - if DDNAME included
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The SUMMARY-BY-DATE Program

SUMMARY-BY-DATE prints a set of 5 summaries of urban sign data. The summaries are:

Summary Number	Summary
A-Ø1	Assemblies by original date (date installed or inventoried)
A-Ø2	Assemblies by most recent date of maintenance
S-Ø1	Signs by original date (date installed, inventoried, or replaced)
S-Ø2	Signs by most recent date of maintenance
S-Ø3	Signs by age

The first four summaries show the most recent ten years broken down by month plus a lump sum total for any dates older than ten years. The last summary, S-Ø3, is based on the sign age as stored for the sign and is shown by year for the most recent twenty years (with a lump sum total of signs older than twenty years).

To use the program, prepare a command that specifies the SUMMARY-BY-DATE program. Command parameters are identical to SUMMARY-BY-CONDITION. The DDNAME parameter is implemented the same manner as in SUMMARY-BY-CONDITION but is not needed since the SUMMARIES parameter can specify up to five summaries.

Figure 1Ø-5 shows a sample output from summary-by-date. The job setup for this output was:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:SUMMARY-BY-DATE,CITY=ALBERTON,SUMMARIES='A-Ø2'
/*
```

SUMMARY-BY-DATE accesses the same files shown above for SUMMARY-BY-CONDITION.

<u>The SUMMARY-BY-SIGN-CODE Program</u>

SUMMARY-BY-SIGN-CODE prints a summary of signs based on sign type (as determined from the sign code). To use the program, prepare a command that specifies the SUMMARY-BY-SIGN-CODE program. Include a CITY parameter (see table 2-1 in chapter 2). You may include the following parameter:

CODE-LIST= $\left\{\begin{array}{l}\text{NO}\\ \underline{\text{YES}}\end{array}\right\}$

CODE-LIST=YES requests a listing of the codes used in the summary (the code list requires one printed page). CODE-LIST=NO is especially useful when receiving output over a CRJE terminal.

You may include any of the following selection parameters (these are described earlier in this chapter in the section "Commands - Urban Sign Subsystem"):

START-ASSEMBLY and/or END-ASSEMBLY
DATE
MAX-#-ENTRIES
SELECT-DD and/or SELECT-SIZE

You may also include any of the print parameters described in chapter 1.

SUMMARY BY DATE CITY OF ALBERTON APR 19,1976

SUMMARY A-02 ASSEMBLIES BY MOST RECENT DATE		INSTALLED (AINS)	INVENTORY (AINV)	REPLACED (ARPL)	REPAIRED (ARPR)	RELOCATED (AREL)	VISIBILTY (AVIS)	CONDITION (ACND)	TOTAL EXISTING	REMOVED (AREM)	TOTAL
1967:	AUG					1			1		1
1968:	DEC	1							1		1
1971:	JAN		1						1		1
1975:	JAN		27						27	1	28
1976:	JAN		5						5		5
	MAR	3	4			1			8		8
	TOTAL	3	9			1			13		13
GRAND TOTAL		4	38			2			43	1	44
1967:	AUG	0.00%	0.00%	0.00%	0.00%	2.27%	0.00%	0.00%	2.27%	0.00%	2.27%
1968:	DEC	2.27%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.27%	0.00%	2.27%
1971:	JAN	0.00%	2.27%	0.00%	0.00%	0.00%	0.00%	0.00%	2.27%	0.00%	2.27%
1975:	JAN	0.00%	61.36%	0.00%	0.00%	0.00%	0.00%	0.00%	61.36%	2.27%	63.64%
1976:	JAN	0.00%	11.36%	0.00%	0.00%	0.00%	0.00%	0.00%	11.36%	0.00%	11.36%
	MAR	6.82%	9.09%	0.00%	0.00%	2.27%	0.00%	0.00%	18.18%	0.00%	18.18%
	TOTAL	6.82%	20.45%	0.00%	0.00%	2.27%	0.00%	0.00%	29.55%	0.00%	29.55%
GRAND TOTAL		9.09%	86.36%	0.00%	0.00%	4.55%	0.00%	0.00%	97.73%	2.27%	100.00%

Figure 1∅-5. SUMMARY-BY-DATE Program.

Figure 1Ø-6 shows a sample output from SUMMARY-BY-SIGN-CODE. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:SUMMARY-BY-SIGN-CODE,CITY=ALBERTON,CODE-LIST=NO
/*
```

SUMMARY-BY-SIGN-CODE accesses the following files:

DD Statement	File
USNxxx	Urban sign file (xxx is city number)
SIGNEDT	Sign code cross-reference file
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages
SORTIN	Scratch file for sort
SORTOUT	Scratch file for sort
SORTWKØ1	Scratch file for sort
SORTWKØ2	Scratch file for sort
SORTWKØ3	Scratch file for sort
SORTLIB	Sort library
SYSOUT	Printed output from sort

Error Messages - Urban Sign Subsystem

***** 38-1ØØ - name PARAMETER MISSING

The indicated required parameter was omitted from the command. Include the parameter on the command and resubmit.

***** 38-1Ø1 - UNKNOWN NAME IN CITY PARAMETER - name

The name coded in the CITY parameter is not a name known to the system. Check table 2-1 in chapter 2 for a list of city names. Correct the command and resubmit.

***** 38-1Ø2 - END-ASSEMBLY PARAMETER /xxxxxx/ SMALLER THAN START-ASSEMBLY PARAMETER /xxxxxx/

If the END-ASSEMBLY parameter is not coded with a value at least as large as START-ASSEMBLY, no assemblies can be selected. Correct the parameter(s) in error and resubmit.

***** 38-1Ø3 - DATE PARAMETER IS IN ERROR - date

The option specified in the DATE parameter is not a valid date in mm/dd/yy format. Correct the DATE parameter and resubmit.

SUMMARY-BY-SIGN-CODE   CITY OF ALBERTON   APR 19,1976

SIGN-CODE	SUPP CODE	DESCRIPTION	DIMENSIONS HORIZ	DIMENSIONS VERT	SHAPE	COLOR	FACE	MATL	LETTER TYPE	NUMBER OF SIGNS	PERCENTAGE OF TOTAL
R03-01-01		DIAGRAM - NO RIGHT TURN	24	24	2	01	3	3	3	1	1.16%
R03-01-11		NO RIGHT TURN	24	18	3	02	3	3	3	1	1.16%
R03-04-01		DIAGRAM - NO U TURN	24	24	2	01	1	4	2	1	1.16%
R03-04-11		NO U TURN	24	18	3	02	3	4	2	1	1.16%
R03-04-11		NO U TURN	24	18	3	02	4	4	3	1	1.16%
R03-04-11		NO U TURN	SIGN CODE SUBTOTAL							2	2.33%
R03-05-01	S	ARROW - STRAIGHT ONLY	30	36	3	02	3	3	4	1	1.16%
R03-05-01	L	ARROW - LEFT AND STRAIGHT ONLY	30	36	3	02	3	3	3	1	1.16%
R04-01-01		DO NOT PASS	24	30	3	02	1	1	2	1	1.16%
R04-01-02		DO NOT PASS	48	60	3	02	1	2	2	1	1.16%
R04-01-02		DO NOT PASS	48	60	3	02	1	2	3	9	10.47%
R04-01-02		DO NOT PASS	SIGN CODE SUBTOTAL							10	11.63%
R04-01-03	**		48	60	3	02	1	2	3	1	1.16%
R04-02-02		PASS WITH CARE	48	60	3	02	1	2	1	1	1.16%
R04-05-01		TRUCKS USE RIGHT LANE	36	44	3	02	1	1	2	1	1.16%
R04-07-01		ARROW - KEEP RIGHT	48	50	3	02	1	1	2	1	1.16%
R04-11-01		ARROW - KEEP LEFT	38	50	4	02	1	1	2	1	1.16%
R04-11-01		ARROW - KEEP LEFT	48	60	3	02	1	1	2	1	1.16%
R04-11-01		ARROW - KEEP LEFT	SIGN CODE SUBTOTAL							2	2.33%
R04-11-11		KEEP LEFT	48	36	3	02	1	1	1	1	1.16%
R04-11-11		KEEP LEFT	48	36	3	02	1	1	2	2	2.33%

Figure 1Ø-6. SUMMARY-BY-SIGN-CODE Program.

***** 38-1Ø4 - UNKNOWN OPTION IN name PARAMETER - FIRST CHARACTER IS x

The option coded in the specified parameter is not a valid option. Check the command format of the program you are running. Correct the option and resubmit.

***** 38-1Ø5 - NO FILE FOR CITY name

No urban sign file has been implemented for the requested city.

Programmer note: This message is caused by a missing USNxxx or USNxxxU DD statement.

***** 38-1Ø6 - NO RECORDS MEET SELECTION CRITERIA

There were no assemblies or signs that met all of the selection criteria specified on the command and/or in the select statement.

***** 38-1Ø7 - WARNING - MAX-#-ENTRIES COMPLETED

This message is a warning that processing was terminated because of a MAX-#-ENTRIES parameter.

***** 38-1Ø8 - START-ASSEMBLY PARAMETER /xxxxxx/ LARGER THAN LAST RECORD IN FILE

The value coded in START-ASSEMBLY is larger than the largest assembly number and tie-breaker stored in the file.

***** 38-1Ø9 - DDNAME PARAMETER MISSING

The required DDNAME parameter was not included on the command.

***** 38-11Ø - BOTH DDNAME AND SUMMARIES SPECIFIED

Either DDNAME or SUMMARIES can be included on the command but both cannot be included. Remove one or the other and resubmit.

***** 38-111 - SUMMARIES PARAMETER IS IN ERROR - option

The SUMMARIES parameter is not written in the correct format. Correct the parameter and resubmit.

***** 38-112 - ABOVE SUMMARY OPTION CARD IS IN ERROR

A summary option card entered via DDNAME is not prepared in the correct format. Correct the card and resubmit.

***** 38-113 - ROUTES /xxxx/ AND /xxxx/ DO NOT INTERSECT

(LIST-SIGNS-BY-STREET program). An AREA data card specifies two route numbers for which no intersection can be found.

***** 38-8ØØ - PROGRAM ERROR IN USNRD SUBRTN - ERROR CODE IS x

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: This message originates in the USNRD subroutine. The error code indicates the cause of the error:

- A Open failed - urban sign file
- B Close failed - urban sign file
- C I/O error - urban sign file
- D USNRDF called twice
- E USNRD called without previous USNRDF call
- F USNRDC called without previous USNRDF call
- G USNRD called after end-of-file
- H Overflow in USNRD

***** 38-8Ø1 - PROGRAM ERROR - RETURN CODE FROM entry-point IS x

***** 38-8Ø1 - PROGRAM ERROR IN xxx SUBRTN - RETURN CODE FROM entry-point IS x

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: This message indicates both the subroutine that returned the unexpected return code and the routine in which the error was detected. The first format of the message is used if the error was detected in the mainline program.

***** 38-8Ø2 - PROGRAM ERROR IN USNINST SUBRTN - DUPLICATE KEY CONDITION RAISED - KEY key

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: When the update program is working right this message cannot be printed, because it checks for a duplicate key before writing a record. The message is the result of (1) a programming error in the USNINST subroutine or (2) a file error - probably allocated with an incorrect key length or relative key position.

***** 38-8Ø3 - PROGRAM ERROR IN USNINST SUBRTN - SIGN RECORD OVERFLOW

More than 26 signs are stored for a given assembly.

***** 38-8Ø4 - PROGRAM ERROR IN xxx SUBRTN - SORT FAILED - RETURN CODE IS x

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: A non-zero return code resulted from a PL/I sort. The sort messages should contain an error message for the problem.

***** 38-8Ø5 - PROGRAM ERROR - EDIT TABLE IN ERROR

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: The error originated in the USNEDRD subroutine and resulted from an error in the edit table. If you specify DEBUG=1 on the command, a listing of the table will be generated and the last record listed is the one in error. If you specify DEBUG=2 on the command, a user-1ØØ abend is issued when the error is detected.

***** 38-8Ø6 - PROGRAM ERROR - EDIT TABLE IN ERROR - ERR-CODE IS xxx

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: The error originated in the USNEDIN subroutine and resulted from an error in the edit table. The error code identifies which record is in error.

***** 38-8Ø7 - PROGRAM ERROR - EDIT TABLE IN ERROR - MESSAGE nnn

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: The error originated in the USNEDER subroutine and resulted from an error in the message portion of the edit table. The message number identifies the record that is in error.

***** 38-8Ø8 - PROGRAM ERROR - subroutine NOT INITIALIZED

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: The indicated subroutine contains an initialization entry point that was not called.

***** 38-8Ø9 - PROGRAM ERROR IN xxx SUBRTN - STORAGE OVERFLOW

This message indicates a programming error. Refer the problem to system maintenance personnel.

Programmer note: The indicated subroutine utilizes a core array for storing the contents of a small file. The file has grown too large for the current core array - increase the size of the array, recompile, and relink.

***** 38-81Ø - PROGRAM ERROR - NO ENTRY IN COORDINATE TABLE

(LIST-SIGNS-BY-STREET program). The coordinate table does not contain an entry for the city being run.

***** 38-811 - PROGRAM ERROR - CODE-LIST PARAMETER CANNOT BE HANDLED

The code listing is generated from a table stored on disk. This message indicates that the table could not be found.

***** 38-812 - PROGRAM ERROR - NO FEET-PER-COORDINATE UNIT SCALES IN COORDINATE TABLE FOR CITY - city

(LIST-SIGNS-BY-STREET program). The entry in the coordinate table for the indicated city does not contain feet-per-coordinate scales.

***** 38-813 - PROGRAM ERROR - ROUTE /xxxx/ DOES NOT EXIST

(LIST-SIGNS-BY-STREET program). A route for which one or more assemblies or signs are stored in the urban sign file does not appear in the city route cross-reference file.

CHAPTER 11

MISCELLANEOUS PROGRAMS

The Grid Table

The grid table is a file containing descriptions of city intersections. A number of cities have now prepared grid tables. The information stored for each intersection includes (1) x- and y-coordinates of the intersection, (2) a verbal description of the intersection, and (3) the city route numbers of the intersecting streets.

The LIST-GRID-TABLE Program - LIST-GRID-TABLE can print the grid table for a specified for city or for all cities. To use the program, prepare a command that specifies the LIST-GRID-TABLE program. If you wish to list the grid table for a specific city, include a CITY parameter (see table 2-1 in chapter 2). If you do not include a CITY parameter, all cities are listed.

Figure 11-1 shows a sample output from LIST-GRID-TABLE. This output is one page obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the data processing Bureau)
// EXEC HISGRID
//SYSIN DD *
:LIST-GRID-TABLE,CITY=BOZEMAN
/*
```

LIST-GRID-TABLE accesses the following files:

DD Statement	File
GRIDTBL	Grid table
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages

The GRIDTBL-SORT-&-LIST Program - GRIDTBL-SORT-&-LIST prints an alphabetical listing of the grid table. Each intersection is printed at least twice - once under each street name. To use the program, prepare a command that specifies the GRIDTBL-SORT-&-LIST program. Include a CITY parameter (see table 2-1 in chapter 2). You may include any of the print parameters described in chapter 1.

HIS GRID TABLE LISTING -- CITY OF BOZEMAN

0681	0258	NORTH 7TH AVE AND BAXTER LANE
0947	0280	NORTH ROUSE (MT 293) AND I-90 OVERPASS
0946	0313	NORTH ROUSE (MT 293) AND EAST OAK ST
0946	0334	NORTH ROUSE (MT 293) AND EAST BIRCH ST
0681	0339	NORTH 7TH AVE AND WEST BIRCH
0682	0390	NORTH 7TH AVE AND WEST JUNIPER
0945	0410	NORTH ROUSE (MT 293) AND EAST TAMARACK
0839	0411	TAMARACK ST AND NORTH TRACY AVE
0869	0411	EAST TAMARACK ST AND NORTH BLACK AVE
0777	0412	WEST TAMARACK ST AND NORTH 3RD AVE
0795	0412	WEST TAMARACK ST AND NORTH GRAND AVE
0632	0415	NORTH 7TH AVE AND WEST TAMARACK
0944	0437	NORTH ROUSE (MT 293) AND EAST ASPEN ST
0632	0440	NORTH 7TH AVE AND WEST ASPEN ST
0944	0463	NORTH ROUSE (MT 293) AND E COTTONWOOD
0816	0463	NORTH WILLSON AVE AND WEST COTTONWOOD ST
1034	0485	EAST PEACH STREET AND NORTH IDA AVE
0971	0486	EAST PEACH ST AND NORTH CHURCH AVE
1061	0486	EAST PEACH ST AND NORTH PLUM AVE
0943	0488	NORTH ROUSE (MT 293) AND EAST PEACH ST
0962	0488	EAST PEACH ST AND NORTH PERKINS PLACE
1004	0488	EAST PEACH ST AND NORTH WALLACE AVE
0489	0489	DURSTON ROAD AND NORTH 15TH AVE
0424	0490	DURSTON ROAD AND NORTH 18TH AVE
0444	0490	DURSTON ROAD AND NORTH 17TH AVE
0896	0490	EAST PEACH ST AND NORTH BOZEMAN AVE
0919	0490	EAST PEACH ST AND NORTH MONTANA AVE
0379	0491	DURSTON ROAD AND NORTH 20TH AVE
0463	0491	DURSTON ROAD AND NORTH 16TH AVE
0588	0491	DURSTON ROAD AND NORTH 11TH AVE
0609	0491	DURSTON ROAD AND NORTH 10TH AVE
0794	0491	WEST PEACH ST AND NORTH GRAND AVE
0817	0491	WEST PEACH ST AND NORTH WILLSON AVE
0843	0491	PEACH ST AND NORTH TRACY AVE
0868	0491	EAST PEACH ST AND NORTH BLACK AVE
0652	0492	DURSTON ROAD AND NORTH 8TH AVE
0725	0492	WEST PEACH ST AND NORTH 5TH AVE
0746	0492	WEST PEACH ST AND NORTH 4TH AVE
0768	0492	WEST PEACH ST AND NORTH 3RD AVE SO-SIDE
0634	0493	DURSTON ROAD AND NORTH 9TH AVE
0682	0494	NORTH 7TH AVE AND WEST PEACH (DURSTON)
0702	0493	WEST PEACH ST AND NORTH 6TH AVE
0817	0517	NORTH WILLSON AVE AND WEST SHORT ST

Figure 11-1. LIST-GRID-TABLE Program.

Figure 11-2 shows a sample output from GRIDTBL-SORT-&-LIST. This figure is one page of the output generated by the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISGRID
//SYSIN DD *
:GRIDTBL-SORT-&-LIST,CITY=BOZEMAN
/*
```

GRIDTBL-SORT-&-LIST accesses the following files:

DD Statement	File
GRIDTBL	Grid table
TABLES	Tables
PRINTER	Printed output (can be changed by PRINTER-DD or OUTPUT-FILE)
SYSPRINT	IBM error messages
DIRTIN	Scratch file for sort
DIRTOUT	Scratch file for sort
DIRTWKØ1	Scratch file for sort
DIRTWKØ2	Scratch file for sort
DIRTWKØ3	Scratch file for sort
SORTLIB	Sort library
SYSOUT	Printed output from sort

The Defense Cross-Reference File

The defense cross-reference file is utilized in producing the defense bridge listings. Maintenance of this file is discussed in the data coding and the system maintenance manuals.

The DEFENSE-XREF Program - DEFENSE-XREF can be used to print a listing of the defense cross-reference file. Prepare a command that specifies the DEFENSE-XREF program and include the parameter TYPE-RUN=LIST or TYPE-RUN=SORT-&-LIST. Figure 11-3 shows a sample output from DEFENSE-XREF. This figure is one of the pages printed by the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISBRID
//SYSIN DD *
:DEFENSE-XREF,TYPE-RUN=LIST
/*
```

GRID TABLE SORT AND LIST -- CITY OF BOZEMAN

X-COORD	Y-COORD	STREET ONE	STREET TWO
1090	0727	EAST CURTISS ST (L-INT)	CYPRESS AVE
0939	0597	EAST LAMME ST	NORTH ROUSE (MT 293)
0872	0653	EAST MAIN (US 191)	BLACK AVE
0898	0653	EAST MAIN (US 191)	BOZEMAN AVE
0971	0653	EAST MAIN (US 191)	CHURCH AVE
1034	0652	EAST MAIN (US 191)	CMSTP&P RR
1235	0697	EAST MAIN (US 191)	HAGGERTY LANE
1047	0652	EAST MAIN (US 191)	NORTH BROADWAY
0939	0653	EAST MAIN (US 191)	ROUSE (MT 293)
1072	0651	EAST MAIN (US 191)	SOUTH BUTTONWOOD
1090	0651	EAST MAIN (US 191)	SOUTH CYPRESS
1004	0652	EAST MAIN (US 191)	WALLACE AVE
0872	0625	EAST MENDENHALL ST	NORTH BLACK AVE
0897	0625	EAST MENDENHALL ST	NORTH BOZEMAN
1052	0625	EAST MENDENHALL ST	NORTH BROADWAY
0969	0623	EAST MENDENHALL ST	NORTH CHURCH AVE
1032	0625	EAST MENDENHALL ST	NORTH PLUM AVE
0939	0625	EAST MENDENHALL ST	NORTH ROUSE AVE
1002	0625	EAST MENDENHALL ST	NORTH WALLACE
0946	0313	EAST OAK ST	NORTH ROUSE (MT 293)
1089	0704	EAST OLIVE ST	CYPRESS AVE
0939	0705	EAST OLIVE ST	SOUTH ROUSE AVE
0868	0491	EAST PEACH ST	NORTH BLACK AVE
0896	0490	EAST PEACH ST	NORTH BOZEMAN AVE
0971	0486	EAST PEACH ST	NORTH CHURCH AVE
0919	0490	EAST PEACH ST	NORTH MONTANA AVE
0962	0488	EAST PEACH ST	NORTH PERKINS PLACE
1061	0486	EAST PEACH ST	NORTH PLUM AVE
0943	0488	EAST PEACH ST	NORTH ROUSE (MT 293)
1004	0488	EAST PEACH ST	NORTH WALLACE AVE
1034	0485	EAST PEACH STREET	NORTH IDA AVE
0945	0410	EAST TAMARACK	NORTH ROUSE (MT 293)
0869	0411	EAST TAMARACK ST	NORTH BLACK AVE
0795	0653	GRAND AVE	WEST MAIN (US 191)
0737	1150	GREEK WAY	KAGY BLVD
1235	0697	HAGGERTY LANE	EAST MAIN (US 191)
1114	0728	HIGHLAND (GOLF WAY)	EAST CURTISS ST
1127	1245	HIGHLAND BLVD	KAGY BLVD
0947	0280	I-90 OVERPASS	NORTH ROUSE (MT 293)
0757	1150	KAGY BLVD	C,M,ST P&P RAILROAD

Figure 11-2. GRIDTBL-SORT-&-LIST Program.

HIS DEFENSE CROSS REFERENCE TABLE

ROUTE_#	START-MILEPOINT	END-MILEPOINT	DEF-SECT	DIR	INVENTORY-RTE	LATITUDE DEG. MIN.	LONGITUDE DEG. MIN.
PC0045	045+0.329	084+0.585	01380	0	21001910		
P00043	000+0.000	042+0.864	01385	0	21001910		
PC0016	220+0.624	311+0.617	01390	0	21001910		
PC0011	000+0.000	052+0.851	01395	0	21000890		
PC0011	052+0.351	053+0.132	01400	0	21000890		
PC0011	060+0.281	060+0.681	01405	0	21000890		
PC0011	060+0.631	117+0.536	01410	0	21000690		
PC0011	129+0.138	200+0.003	01415	0	21000890		
P00011	223+0.540	225+0.717	01420	0	21000890		
IC0015	399+0.771	400+0.397	01430	X	21000890		
I00015	399+0.571	399+0.771	01435	X	21000890		
PC0003	299+0.631	307+0.105	01445	0	21000890		
PC0003	307+0.105	340+0.214	01450	0	21000890		
PC0003	340+0.214	409+0.633	01455	0	21000890		
PC0003	413+0.039	425+0.053	01460	0	21000890	48 33.0	113 10.0
PC0003	425+0.053	463+0.366	01465	0	21000690		
SC0357	000+0.000	010+0.523	01470	X	81003970		
SC0308	014+0.582	025+0.467	01475	0	81003080		
P00004	000+0.000	024+0.736	01480	0	21003100		
PC0004	024+0.736	042+0.765	01485	0	21003100		
PC0016	103+0.910	104+0.307	01490	0	21000870		
PC0015	104+0.307	152+0.022	01495	0	21000870		
PC0016	154+0.219	199+0.150	01510	0	21000870		
PC0015	082+0.560	112+0.982	01515	X	21000870		
P00015	064+0.958	082+0.056	01520	X	21000870		
PC0015	000+0.000	064+0.958	01525	X	21000870		
PC0011	200+0.003	223+0.540	01530	0	21000870		
P00010	000+0.000	089+0.418	01535	0	21000870		
PC0010	089+0.418	111+0.369	01540	0	21000870		
PC0054	000+0.000	029+0.435	01545	0	31000590	45 10.0	105 21.3
PC0049	000+0.335	027+0.612	01550	0	31000410		
PC0029	042+0.344	066+0.000	01555	0	31000410		

Figure 11-3. DEFENSE-XREF Program.

The City Route Cross-Reference Files

The location method used in the urban sign inventory has resulted in the need for assigning a unique number to each city street in those cities using the urban sign software. Each city has its own separate city route cross-reference file that contains an inventory of these numbers.

The CITY-ROUTE-XREF Program - CITY-ROUTE-XREF can be used to list a city route cross-reference file. Prepare a command that specifies the CITY-ROUTE-XREF program and include a CITY parameter (see table 2-1 in chapter 2). Include the parameter TYPE-RUN=LIST. You may include any of the print parameters described in chapter 1. Figure 11-4 is a sample output from CITY-ROUTE-XREF. This output was obtained with the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:CITY-ROUTE-XREF,CITY=ALBERTON,TYPE-RUN=LIST
/*
```

The Sign Code Cross-Reference Files

The sign code cross-reference files contain an inventory of all the known sign code numbers. There are two of these files: one based on the federal sign codes and one based on the Montana sign codes. Both files are maintained by the SIGN-CODE-XREF program.

The SIGN-CODE-XREF Program - SIGN-CODE-XREF can be used to list the sign code cross-reference files. Prepare a command that specifies the SIGN-CODE-XREF program and include the parameter TYPE-RUN=LIST. Figure 11-5 shows a sample output from the SIGN-CODE-XREF program. This output is one page of the output resulting from the following job setup:

```
// JOB    (a JOB card is obtained from the Data Processing Bureau)
// EXEC HISUSN
//SYSIN DD *
:SIGN-CODE-XREF,TYPE-RUN=LIST
/*
```

CITY-ROUTE-XREF APR 19,1976

CITY OF ALBERTON

ROUTE-NO	STREET NAME	A-DIREC	B-DIREC
0101	1ST AVENUE	180	000
0102	2ND AVENUE		000
0103	3RD AVENUE	180	000
0104	4TH AVENUE	180	000
0105	5TH AVENUE	180	000
0777	ANGLE BOULEVARD	225	045
0888	CURVE DRIVE	270	090
0901	A STREET	270	090
0902	B STREET	270	090
0903	C STREET	270	
0904	D STREET	270	090
0905	E STREET	270	090

*** END OF FILE ***

Figure 11-4. CITY-ROUTE-XREF Program.

SIGN-CODE-XREF APR 15,1976 TYPE-RUN=LIST

MONTANA CODE	SUPP CODE	FEDERAL CODE	DIMENSIONS HORZ	VERT	COLORS	SHAPE	REMARKS
R04-12-01		R4-12	48	60	02-BLACK AND WHITE	3-RECTANGLE	NO DRIVING ON SHOULDER
R05-01-01		R5-1	30	30	01-RED AND WHITE	2-SQUARE	DO NOT ENTER
R05-01-02		R5-1	36	36	01-RED AND WHITE	2-SQUARE	DO NOT ENTER
R05-02-01		R5-2	24	24	01-RED AND WHITE	2-SQUARE	DIAGRAM - NO TRUCKS
R05-02-11		R5-2P	24	18	02-BLACK AND WHITE	3-RECTANGLE	NO TRUCKS
R05-03-01		R5-3	24	12	02-BLACK AND WHITE	3-RECTANGLE	PEDESTRIANS PROHIBITED
R05-04-01		R5-4	24	30	02-BLACK AND WHITE	3-RECTANGLE	COMMERCIAL VEHICLES EXCLUDED
R05-05-01		R5-5	24	30	02-BLACK AND WHITE	3-RECTANGLE	VEHICLES WITH LUGS PROHIBITED
R05-06-01		R5-6	24	24	01-RED AND WHITE	2-SQUARE	DIAGRAM - NO BICYCLES
R05-06-11		R5-6P	24	18	02-BLACK AND WHITE	3-RECTANGLE	NO BICYCLES
R05-09-01		R5-9	36	24	01-RED AND WHITE	3-RECTANGLE	WRONG WAY
R05-10-01		R5-10	30	36	02-BLACK AND WHITE	3-RECTANGLE	PEDESTRIANS, BIKES PROHIBITED
R05-10-02		R5-10	30	18	02-BLACK AND WHITE	3-RECTANGLE	PEDESTRIANS & BIKES PROHIBITED
R06-01-01		R6-1L	36	12	02-BLACK AND WHITE	3-RECTANGLE	ONE WAY LEFT
R06-01-02		R6-1R	36	12	02-BLACK AND WHITE	3-RECTANGLE	ONE WAY RIGHT
R06-02-01		R6-2L	18	24	02-BLACK AND WHITE	3-RECTANGLE	ONE WAY LEFT (LIMITED SPACE)
R06-02-02		R6-2R	18	24	02-BLACK AND WHITE	3-RECTANGLE	ONE WAY RIGHT (LIMITED SPACE)
R07-01-01		R7-1	12	18	01-RED AND WHITE	3-RECTANGLE	NO PARKING ANY TIME
R07-02-01		R7-2	12	18	01-RED AND WHITE	3-RECTANGLE	NO PARKING XX AM TO XXPM
R07-03-01		R7-3	12	18	01-RED AND WHITE	3-RECTANGLE	NO PARKING EXCEPT SUNDAYS & HOLIDAYS
R07-04-01		R7-4	12	18	01-RED AND WHITE	3-RECTANGLE	NO STOPPING OR STANDING
R07-05-01		R7-5	12	18	03-GREEN AND WHITE	3-RECTANGLE	HOUR PARKING XX AM TO XX PM
R07-06-01		R7-6	12	18	01-RED AND WHITE	3-RECTANGLE	NO PARKING - LOADING ZONE
R07-07-01		R7-7	12	18	01-RED AND WHITE	3-RECTANGLE	NO PARKING - BUS STOP
R07-107-01		R7-107	12	18	01-RED AND WHITE	3-RECTANGLE	NO PARKING - BUS STOP
R07-108-01		R7-108	12	18	03-GREEN AND WHITE	3-RECTANGLE	2 HOUR PARKING - ## AM TO ## PM
R07-201-01		R7-201	12	6	01-RED AND WHITE	3-RECTANGLE	TOW-AWAY ZONE
R08-01-01		R8-1	24	30	01-RED AND WHITE	3-RECTANGLE	NO PARKING ON PAVEMENT
R08-02-01		R8-2	24	30	01-RED AND WHITE	3-RECTANGLE	NO PARKING EXCEPT ON SHOULDER
R08-03-01		R8-3	24	30	01-RED AND WHITE	3-RECTANGLE	NO PARKING
R08-04-01		R8-4	30	24	02-BLACK AND WHITE	3-RECTANGLE	EMERGENCY PARKING ONLY
R08-05-01		R8-5	24	30	01-RED AND WHITE	3-RECTANGLE	NO STOPPING ON PAVEMENT
R08-06-01		R8-6	24	30	01-RED AND WHITE	3-RECTANGLE	NO STOPPING EXCEPT ON SHOULDER
R08-07-01		R8-7	30	24	02-BLACK AND WHITE	3-RECTANGLE	EMERGENCY STOPPING ONLY
R09-01-01		R9-1	18	24	02-BLACK AND WHITE	3-RECTANGLE	WALK ON LEFT FACING TRAFFIC
R09-02-01		R9-2	12	18	02-BLACK AND WHITE	3-RECTANGLE	CROSS ONLY AT CROSSWALKS
R09-03-01		R9-3	12	18	02-BLACK AND WHITE	3-RECTANGLE	NO PEDESTRIAN CROSSING
R09-04-01		R9-4	18	24	02-BLACK AND WHITE	3-RECTANGLE	NO HITCH HIKING

Figure 11-5. SIGN-CODE-XREF Program.

CHAPTER 12

THE SELECT SUBSYSTEM

Introduction

The select subsystem is a major enhancement that has been added to the Highway Information System for release 4.∅. Release 3.∅ contained select routines in the accident subsystem. During the implementation of release 4.∅, the release 3.∅ accident select routines were eliminated and a set of system-wide routines installed.

The select routines provide the capability of "selecting," or singling out, records for processing by a summary-generating program. For example, when listing the roadlog file it is possible to include in the listing only those roadway sections that were built before 1945. Note that the select routines do not provide the capability of generating a summary in user-defined format.

Identifying Data Elements

The select routines provide the capability of selecting records based on the contents of any of the data elements in the records (and in many cases, based on the contents of any of the data elements in the records of other files). To allow users an easy-to-use method of specifying selection criteria, each data element of each file is assigned a data element name. The name consists of a 3-character file identifier followed by a period and a field name. For example, the year built field of the roadlog file has the name RLG.YEAR-BUILT. Appendix B contains a complete list of the data element names and the field contents.

Data Element Relationships

Relationships are used to specify selection criteria. Two formats of relationships can be used. The first type compares a data element with a constant value, as in the relationship RLG.YEAR-BUILT *EQ* 55 (this relationship selects only those roadlog sections that were built during 1955). The second type compares two data elements, as in the relationship RLG.YEAR-BUILT *EQ* RLG.YEAR-IMPROVED (this relationship selects only those roadlog sections in which the year built and the year improved are identical).

As can be seen, each relationship consists of three parts: 1) a data element name, 2) a comparison symbol, and 3) a value or data element for comparison. The following comparison symbols can be used:

EQ Equal
NE Not equal
GT Greater than
GE Greater than or equal
LT Less than
LE Less than or equal

Examples:

RLG.YEAR-BUILT *GE* 55 (year built 1955 or more recent)

RLG.YEAR-IMPROVED *LT* RLG.YEAR-BUILT (year improved older than year built)

When preparing a relationship, you must know (1) the format in which the data elements are stored and (2) what the contents of the data elements are. Appendix B contains all of this information for all of the data elements in the system. The data element format includes (1) type of field, (2) length of field, and (3) if decimal, the number of digits to the right of the decimal point. The field types are:

1. Character.
2. Decimal.
3. Date.

The field length is the total number of characters (character format), the total number of digits (decimal format), or 6 (date format). Some decimal fields contain an imbedded decimal point. For example, the section length field of the roadlog file has a length of 5 digits with 3 digits to the right of the decimal point. The data element contains a value that ranges from Ø to 99.999.

Relationships With Two Data Elements - When you prepare a relationship that compares two data elements (as in RLG.YEAR-BUILT *EQ* RLG.YEAR-IMPROVED), the two data elements must be stored in identical format. They must have the same type (character, decimal, or date), the same length, and (if decimal) the same number of digits to the right of the decimal point.

Relationships With One Data Element - When you prepare a relationship that compares a data element to a constant (as in RLG.YEAR-BUILT *EQ* 55), you must be aware of the data element type (character, decimal, or date). The constants are written in different formats for the three types. A

character constant is written within single quotes, as in RLG.REMARK *EQ* 'EN'. The constant need not be of the same length as the data element. If the constant has fewer characters than the data element length, it is treated as left-justified and is padded on the right with blanks. If the constant has more characters than the data element length, the leftmost characters are used and the additional characters on the right are ignored.

A decimal constant is written simply as a number. It can be preceded by a sign (+ or -), and it can contain a decimal point. The constant need not have the same length as the data element. If the constant is shorter than the data element (as in RLG.YEAR-BUILT *EQ* 5), it is treated as right-justified with leading zeroes (eg., Ø5). If the constant is longer than the data element (as in RLG.YEAR-BUILT *EQ* 1955), the higher-level digits are ignored (this value is treated as 955 because all decimal data elements have an odd number of digits). Examples of decimal relationships are:

```
RLG.SECTION-LENGTH *GE* 1
RLG.SECTION-LENGTH *EQ* 1.534
RLG.YEAR-BUILT *LE* 48
RRX.GRADE-1 *LE* -1
```

A data constant is written in the format mm/dd/yy, where "mm" is the month, "dd" is the day, and "yy" is the year. Leading zeroes are accepted but are not needed. Examples are:

```
RLG.EFFECTIVE-DATE *GE* 1/1/74
RLG.EFFECTIVE-DATE *EQ* 12/25/76
```

Select Statements

When submitting a run in which you want to use the select subsystem, you must prepare a select statement. Each select statement consists of one or more relationships as described above. When more than one relationship is needed, they are separated by "connectors." The connectors are *OR* and *AND*.

An example of a select statement containing two relationships is:

```
RLG.YEAR-BUILT *EQ* 45  *OR*  RLG.YEAR-IMPROVED *EQ* 45
```

This statement is interpreted to select all roadlog sections in which either the year built is 45 or the year improved is 45. If the connector *OR* is replaced with *AND*, the statement is interpreted to select roadlog sections in which <u>both</u> the year built is 45 <u>and</u> the year improved is 45.

The connector *AND* is given a higher "priority" than the connector *OR*. For example, the relationship

```
RLG.YEAR-BUILT *EQ* 45  *AND*  RLG.YEAR-IMPROVED *EQ* 45  *OR*
RLG.YEAR-BUILT *EQ* 5Ø  *AND*  RLG.YEAR-IMPROVED *EQ* 5Ø
```

selects those records that have both years equal to 45 as well as those records that have both years equal to 5Ø.

Parenthesis can be used to influence the priority of connectors. For example, the relationship

```
RLG.YEAR-BUILT *EQ* 45  *AND*
(RLG.YEAR-IMPROVED *EQ* 45  *OR*  RLG.YEAR-IMPROVED *EQ* 46)
```

selects those sections in which the year built is 45 <u>and</u> the year improved is either 45 or 46. Note that this is considerably different from the statement

```
RLG.YEAR-BUILT *EQ* 45 *AND* RLG.YEAR-IMPROVED *EQ* 45  *OR*
RLG.YEAR-IMPROVED *EQ* 46
```

which selects those sections in which both years are 45 plus those sections in which the year improved is 46 (regardless of the year built).

<u>Coding Select Statements</u>

Select statements are coded in free-form. Any number of cards can be used for a single select statement. Blanks may be used freely to separate items. Any of columns 1-72 of the cards can be used. As an example of free form, the select statement

```
RLG.YEAR-BUILT *EQ* 11  *AND*  RLG.YEAR-IMPROVED *LT* 2Ø
```

is exactly equivalent to the statement

```
RLG.YEAR-BUILT    *EQ*    1       *AND*
  RLG.YEAR-IMPROVED *LT*   2Ø
```

The SELECT-DD Parameter

The SELECT-DD parameter serves as a linkage between a command and a select statement. The job setup when using select is:

```
// JOB
// EXEC catproc
//SYSIN DD *
:prognam,...,SELECT-DD=name,...
/*
//name DD *
   select statement
/*
```

An example of a job setup is:

```
// JOB
// EXEC HISACC
//SYSIN DD *
:LIST,FILE=ACCIDENT,CITY=GREAT-FALLS,SELECT-DD=SELECTA
/*
//SELECTA DD *
    DAC.#-FATALITIES *GT* Ø
/*
```

In the SELECT-DD parameter, code a 7 or 8 character name that begins with the characters SELECT. Use the same name on a "DD *" DD statement and place the select statement after this DD statement.

The SELECT-SIZE Parameter

The system has a built-in limitation on the size of a select statement. In most cases, the size limitation is large enough. However, a very lengthy select statement may exceed the limitation. SELECT-SIZE can be used to increase the system's limitation. SELECT-SIZE=1 is the default. SELECT-SIZE=2 doubles the maximum statement size. SELECT-SIZE=3 triples the maximum size. The largest value that can be specified is SELECT-SIZE=9.

Efficiency

Any select statement that has two or more relationships can be written in two or more ways. For example, the relationship

```
RLG.YEAR-BUILT *EQ* 45  *OR*  RLG.YEAR-BUILT *EQ* 46
```

can also be written as

```
RLG.YEAR-BUILT *EQ* 46  *OR*  RLG.YEAR-BUILT *EQ* 45
```

The efficiency of the run can be seriously affected by the manner in which the statement is written. A run that might cost $2Ø could cost $2ØØ or more if the select statement is badly written.

Each relationship specifies a test that the software must perform. The relationship RLG.YEAR-BUILT *EQ* 45 specifies that the software must test each roadlog record for the presence of a 45 in the year built field. When two or more relationships are coded, the software performs the tests in the order in which they are coded and only performs tests that are needed. Consider the following relationship:

```
RLG.YEAR-BUILT *EQ* 45  *OR*  RLG.YEAR-BUILT *EQ* 46
```

If a record is read that contains a 45 in the year built field, only the first test has to be made. If a record is read that contains anything else in the year built field, both tests have to be made. Similarly, in the relationship

```
RLG.YEAR-BUILT *EQ* 45  *AND*  RLG.YEAR-IMPROVED *EQ* 5Ø
```

The second test is performed only when a record is read that contains a 45 in the year built field.

In the cases discussed so far, there is very little difference in efficiency when the statement is rearranged. Efficiency becomes very important, however, when a statement refers to more than one file. It is important to identify which file is accessed first and to arrange the select statement to refer to that file first. For example, consider the accident LIST program. You wish to select only those accidents involving a motorcycle in which one or more persons were killed. The

select statement can be written either as

VAC.BODY-STYLE *EQ* 8 *AND* DAC.#-FATALITIES *NE* Ø

or as

DAC.#-FATALITIES *NE* Ø *AND* VAC.BODY-STYLE *EQ* 8

The accident LIST program reads the accident detail file to produce its listing. If the select statement is written in the first manner, the accident vehicle file is accessed once for each accident and the second relationship is tested only for accidents involving motorcycles. If the command specifies a one-year period, the vehicle file is accessed approximately 1Ø,ØØØ times. If the statement is written the second way, the vehicle file is accessed only for fatal accidents. The number of accesses drops to about 2ØØ. This represents a considerable savings because direct file accesses represents the single highest cost in running Highway Information System Programs.

Each program implemented with select has a "major" file and may have one or more "minor" files. The major file is the file that drives the program (the accident detail file in the above example). The minor files are any other files that can be selected on. Always code the select statement to refer to the major file first. The major file depends upon which subsystem you are using.

Accident Subsystem - If the command does not contain a DATA parameter, the major file is the accident detail file and the minor file is the accident vehicle file. If the command includes a DATA parameter, the major file is the accident directory file and the following files are minor files: roadlog, traffic, accident detail, and accident vehicle.

Bridge Subsystem - The bridge file is the major file. The roadlog and traffic files are minor files.

Railroad Subsystem - The railroad file is the major file. The roadlog and traffic files are minor files.

Roadlog Subsystem - The roadlog file is the major file. The traffic file is a minor file.

The Skid Subsystem - The skid file is the major file. The roadlog and traffic files are minor files but cannot be referred to if a CITY parameter is included on the command.

The Sufficiency Subsystem - The sufficiency file is the major file. The roadlog and traffic files are minor files.

The Traffic Subsystem - The traffic file is the major file. The roadlog file is a minor file.

The Urban Sign Subsystem - The urban sign file is the major file. There are no minor files.

Error Messages

***** SELECT-Ø1 -- DATA ELEMENT NAME TOO LONG: name

Data element names consist of a 3-character file name, a period, and a field name that can be up to 2Ø characters in length. This message is printed if a name is coded that exceeds 24 characters. This message may also be printed if a character constant longer than 24 characters is not enclosed in the required quotes.

***** SELECT-Ø2 -- UNMATCHED QUOTE

The ending quote of a character constant was omitted.

***** SELECT-Ø3 -- CHARACTER STRING TOO LONG: string

The maximum length of a character constant is 35 characters.

***** SELECT-Ø4 -- INVALID DATE

An invalid date was coded in a date constant.

***** SELECT-Ø5 -- UNKNOWN COMPARISON SYMBOL

A comparison symbol other than *EQ*,*NE*,*GE*,*GT*,*LE*, or *LT* was coded.

***** SELECT-Ø6 -- UNBALANCED PARENTHESIS

The statement contains an unequal number of left and right parenthesis.

***** SELECT-11 -- DATA ELEMENT FORMAT ERROR: name

The fourth character of a data element name is not a period.

***** SELECT-12 -- FILE NAME UNKNOWN: name

The first three characters of a data element name does not correspond to one of the implemented files.

***** SELECT-13 -- DATA ELEMENT UNKNOWN: name

The data element name is not one known to the select software.

***** SELECT-21 -- INVALID CONSTANT SPECIFIED FOR name

The constant specified for the data element is not in the proper format (eg., the constant 3Ø specified for a character-format data element).

***** SELECT-22 -- DATA ELEMENTS HAVE DIFFERENT ATTRIBUTES: name1,name2

A relationship specified a comparison between two data elements that have different formats (eg., one is character and the other decimal) or different lengths.

***** SELECT-23 -- UNKNOWN CITY FOR name -- CITY=city

An unknown city name was specified in a data element that contains a city number.

***** SELECT-24 -- UNKNOWN COUNTY FOR name -- COUNTY=county

An unknown county name was specified in a data element that contains a county number.

***** SELECT-25 -- UNKNOWN PROJECT CLASS FOR name -- CLASS=class

An unknown project classification was specified.

***** SELECT-51 -- STORAGE OVERFLOW -- SPECIFY LARGER SELECT-SIZE AND RESUBMIT

The select statement is too large to run as is. Include a SELECT-SIZE parameter to allow a larger statement.

***** SELECT-52 -- SYNTAX ERROR

The select statement contains an invalid combination of items such as two data element names in succession without an intervening comparison symbol or connector symbol.

***** SELECT-91 -- ERROR IN SELECT BLOCK

This message indicates a programming error. Refer the problem to system maintenance personnel.

***** SELECT-92 -- FILE NAME DOES NOT EXIST

This message indicates a programming error. Refer the problem to system maintenance personnel.

***** SELECT-1ØØ -- ERROR IN READING MINOR FILE

This message indicates a programming error or a file error. Refer the problem to system maintenance personnel.

***** SELECT-1Ø1 -- MINOR FILE NOT IMPLEMENTED

The select statement refers to a file that cannot be accessed by the program being run (eg., ACC.ACCIDENT-NUMBER specified in a select statement prepared for the roadlog LIST program).

***** SELECT-12Ø -- DATA PARAMETER MISSING

(Accident subsystem only). The roadlog, traffic, and accident directory file cannot be selected on unless a DATA parameter is included on the command.

***** SELECT-121 -- NO DETAIL RECORD FOR ACCIDENT number

This message indicates an error in the accident files. Refer the problem to system maintenance personnel.

***** SELECT-122 -- NO VEHICLE RECORD FOR ACCIDENT number

This message indicates an error in the accident files. Refer the problem to system maintenance personnel.

***** SELECT-131 -- UNIMPLEMENTED SPECIAL OPTION

This message indicates a programming error. Refer the problem to system maintenance personnel.

APPENDIX A

CATALOGED PROCEDURES

This appendix contains listings of all of the HIS cataloged procedures. The procedures are listed in the following order:

----- LISTPDS -----

LIBRARY=RUNLIB   MEMBER=HIS

```
//*    -----> HIS PROCEDURE <-----
//        PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//             SCR1=1,SCR2=1,OUTLIM=20000
//HIS EXEC PGM=HIS20400
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//SORTLIB DD DISP=SHR,DSN=SYS1.SORTLIB
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=&DISP,DSNAME=HIS.TABLES
//SORTIN   DD UNIT=SYSDA,SPACE=(CYL,&SCR2)
//SORTOUT  DD UNIT=SYSDA,DISP=(OLD,PASS),VOL=REF=*.SORTIN,
//            DSNAME=*.SORTIN
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&SCR1)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&SCR1)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&SCR1)
//SCRATCH1 DD UNIT=SYSDA,DISP=(OLD,PASS),VOL=REF=*.SORTWK01,
//            DSNAME=*.SORTWK01
//SCRATCH2 DD UNIT=SYSDA,DISP=(OLD,PASS),VOL=REF=*.SORTWK02,
//            DSNAME=*.SORTWK02
//SCRATCH3 DD UNIT=SYSDA,DISP=(OLD,PASS),VOL=REF=*.SORTWK03,
//            DSNAME=*.SORTWK03
```

LIBRARY=RUNLIB   MEMBER= HISACC

```
//*    -----> HISACC PROCEDURE <-----
//        PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//             OUTLIM=20000,SCR1=1
//HIS EXEC PGM=HIS20400
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
```

----- LISTPDS -----

```
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG,UNIT=(,,DEFER)
//TRUMILE  DD DISP=SHR,DSNAME=HIS.TRUMILE,UNIT=(,,DEFER)
//ACIDENT  DD DISP=&DISP,DSNAME=HIS.ACCIDENT
//ACCVEH   DD DISP=&DISP,DSNAME=HIS.ACCVEH
//ACCDIRI  DD DISP=&DISP,DSNAME=HIS.ACCDIRI
//SCRATCH1 DD UNIT=SYSDA,SPACE=(CYL,&SCR1)
//SCRATCH2 DD UNIT=SYSDA,SPACE=(CYL,&SCR1)
```

LIBRARY=RUNLIB   MEMBER=HISACCA

```
//*     -----> HISACCA PROCEDURE <-----
//      PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//           OUTLIM=20000,SCR1=1
//HIS EXEC PGM=HIS20400
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//SORTLIB  DD DISP=SHR,DSNAME=SYS1.SORTLIB
//SCRATCH1 DD UNIT=SYSDA,SPACE=(CYL,&SCR1)
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG,UNIT=(,,DEFER)
//TRUMILE  DD DISP=SHR,DSNAME=HIS.TRUMILE
//TRAFFIC  DD DISP=SHR,DSNAME=HIS.TRAFFIC
//ACIDENT  DD DISP=SHR,DSNAME=HIS.ACCIDENT
//ACCVEH   DD DISP=SHR,DSNAME=HIS.ACCVEH
//ACCDIRI  DD DISP=SHR,DSNAME=HIS.ACCDIRI
//ACCSECT  DD DISP=&DISP,DSNAME=HIS.ACCSECT
```

----- LISTPDS -----

LIBRARY=RUNLIB   MEMBER=HISACCM

```
//*    -----> HISACCM PROCEDURE <-----
//         PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//              OUTLIM=20000,
//              DIRSIZE=5,INDSIZE=1,SCR4=1
//HIS EXEC PGM=HIS20400
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SORTLIB   DD DISP=SHR,DSNAME=SYS1.SORTLIB
//INSTRCT   DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3    DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1    DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE      DD SYSOUT=Z,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP   DD SYSOUT=A
//SYSOUT    DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES    DD DISP=SHR,DSNAME=HIS.TABLES
//ACCIDENT  DD DISP=SHR,DSNAME=HIS.ACCIDENT
//ACCVEH    DD DISP=SHR,DSNAME=HIS.ACCVEH
//DIRECT    DD UNIT=SYSDA,SPACE=(CYL,(&DIRSIZE,,1)),DCB=(BLKSIZE=1000,
//             LRECL=50,RECFM=FB,KEYLEN=16,RKP=1,OPTCD=LYR,DSORG=IS)
//GRIDTBL   DD DISP=&DISP,DSNAME=HIS.GRIDTBL
//DIRTIN    DD UNIT=SYSDA,SPACE=(CYL,&DIRSIZE)
//DIRTOUT   DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.DIRTIN,DSNAME=*.DIRTIN
//DIRTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//DIRTWK02 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//DIRTWK03 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//INDXIN    DD UNIT=SYSDA,SPACE=(CYL,&INDSIZE)
//INDXOUT   DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.INDXIN,DSNAME=*.INDXIN
//INDXWK01 DD DISP=OLD,DSNAME=*.DIRTWK01,VOL=REF=*.DIRTWK01
//INDXWK02 DD DISP=OLD,DSNAME=*.DIRTWK02,VOL=REF=*.DIRTWK02
//INDXWK03 DD DISP=OLD,DSNAME=*.DIRTWK03,VOL=REF=*.DIRTWK03
//SCRATCH1 DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.DIRTWK01,DSNAME=*.DIRTWKO
//SCRATCH2 DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.DIRTWK02,DSNAME=*.DIRTWKO
//SCRATCH3 DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.DIRTWK03,DSNAME=*.DIRTW
//SCRATCH4 DD UNIT=SYSDA,SPACE=(CYL,&SCR4)
```

----- LISTPDS -----

```
LIBRARY=RUNLIB          MEMBER=HISACD

//*     -----> HISACD PROCEDURE <-----
//      PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//           OUTLIM=20000
//HIS EXEC PGM=HIS20400
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT   DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3    DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1    DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE      DD SYSOUT=Z,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP   DD SYSOUT=A
//SYSOUT    DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES    DD DISP=SHR,DSNAME=HIS.TABLES
//SORTLIB   DD DISP=SHR,DSNAME=SYS1.SORTLIB
//DIRTIN    DD UNIT=SYSDA,SPACE=(CYL,(10,3))
//DIRTOUT   DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.DIRTIN,DSNAME=*.DIRTIN
//DIRTWK01 DD UNIT=SYSDA,SPACE=(CYL,(6),,CONTIG)
//DIRTWK02 DD UNIT=SYSDA,SPACE=(CYL,(6),,CONTIG)
//DIRTWK03 DD UNIT=SYSDA,SPACE=(CYL,(6),,CONTIG)
//ROADLOG   DD DISP=SHR,DSNAME=HIS.ROADLOG,UNIT=(,,DEFER)
//ACCIDENT  DD DISP=SHR,DSNAME=HIS.ACCIDENT
//ACCVEH    DD DISP=SHR,DSNAME=HIS.ACCVEH
//ACCDIRI   DD DISP=&DISP,DSNAME=HIS.ACCDIRI
```

```
LIBRARY=RUNLIB          MEMBER=HISBRID

//*     -----> HISBRID PROCEDURE <-----
//      PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//           OUTLIM=20000
//HIS EXEC PGM=HIS20400
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT   DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3    DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1    DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE      DD SYSOUT=Z,OUTLIM=&OUTLIM,
```

```
//                  DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//DEFENSE  DD DISP=&DISP,DSNAME=HIS.DEFENSE
//DEFENSEI DD DISP=SHR,DSNAME=HIS.DEFENSE
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG
//TRUMILE  DD DISP=SHR,DSNAME=HIS.TRUMILE
//TRAFFIC  DD DISP=SHR,DSNAME=HIS.TRAFFIC
//BRIDGE   DD DISP=&DISP,DSNAME=HIS.BRIDGE
//BDGREP   DD DISP=SHR,DSNAME=HIS.BDGREP
//SORTLIB  DD DISP=SHR,DSNAME=SYS1.SORTLIB
//SORTIN   DD DISP=SHR,DSNAME=HIS.BDGREP
//SORTOUT  DD DISP=SHR,DSNAME=HIS.BDGREP
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//BDGFILE  DD UNIT=SYSDA,SPACE=(CYL,(7,,1)),
//                  DCB=(LRECL=122,RECFM=FB,BLKSIZE=2440,
//                  KEYLEN=16,RKP=1,DSORG=IS)
//SRTLIN   DD UNIT=SYSDA,SPACE=(CYL,1)
//SRTLOUT  DD UNIT=SYSDA,SPACE=(CYL,1)
//SRTLWK01 DD UNIT=SYSDA,DISP=(OLD,PASS),DSNAME=*.SORTWK01,
//                  VOL=REF=*.SORTWK01
//SRTLWK02 DD UNIT=SYSDA,DISP=(OLD,PASS),DSNAME=*.SORTWK02,
//                  VOL=REF=*.SORTWK02
//SRTLWK03 DD UNIT=SYSDA,DISP=(OLD,PASS),DSNAME=*.SORTWK03,
//                  VOL=REF=*.SORTWK03
```

LIBRARY=PUNLIB   MEMBER=HISGRID

```
//*    -----> HISGRID PROCEDURE <-----
//        PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//             OUTLIM=20000,DIRSIZE=5,INDSIZE=1
//HIS  EXEC PGM=HIS20400
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SORTLIB   DD DISP=SHR,DSNAME=SYS1.SORTLIB
//INSTRCT   DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//                DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3    DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//                DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1    DD SYSOUT=(C,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//                DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE      DD SYSOUT=Z,OUTLIM=&OUTLIM,
```

----- LISTPDS -----

```
//                  DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT  DD  SYSOUT=A
//PLIDUMP   DD  SYSOUT=A
//SYSOUT    DD  SYSOUT=A
//SYSUDUMP  DD  SYSOUT=A
//TABLES    DD  DISP=SHR,DSNAME=HIS.TABLES
//SGNXREF   DD  DISP=&DISP,DSNAME=HIS.SGNXREF
//SIGNEDT   DD  DISP=&DISP,DSNAME=HIS.SIGNEDT
//SAVESXR   DD  DISP=OLD,DSNAME=HIS.SAVESXR
//GRIDTBL   DD  DISP=&DISP,DSNAME=HIS.GRIDTBL
//DIRTIN    DD  UNIT=SYSDA,SPACE=(CYL,&DIRSIZE)
//DIRTOUT   DD  UNIT=SYSDA,DISP=OLD,VOL=REF=*.DIRTIN,DSNAME=*.DIRTIN
//DIRTWK01  DD  UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//DIRTWK02  DD  UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//DIRTWK03  DD  UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//INDXIN    DD  UNIT=SYSDA,SPACE=(CYL,&INDSIZE)
//INDXOUT   DD  UNIT=SYSDA,DISP=OLD,VOL=REF=*.INDXIN,DSNAME=*.INDXIN
//INDXWK01  DD  DISP=OLD,DSNAME=*.DIRTWK01,VOL=REF=*.DIRTWK01
//INDXWK02  DD  DISP=OLD,DSNAME=*.DIRTWK02,VOL=REF=*.DIRTWK02
//INDXWK03  DD  DISP=OLD,DSNAME=*.DIRTWK03,VOL=REF=*.DIRTWK03
//SCRATCH1  DD  DISP=OLD,DSNAME=*.DIRTWK01,VOL=REF=*.DIRTWK01
```

LIBRARY=RUNLIB   MEMBER=HISMEMO

```
//*    -----> HISMEMO PROCEDURE <-----
//     PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//          OUTLIM=20000
//HIS  EXEC PGM=HIS20400
//STEPLIB   DD  DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT   DD  UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD  SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//              DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//PRNTMEMO  DD  SYSOUT=(D,,2461),COPIES=&COPIES,OUTLIM=&OUTLIM,
//              DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//CRJE      DD  SYSOUT=Z,OUTLIM=&OUTLIM,
//              DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT  DD  SYSOUT=A
//PLIDUMP   DD  SYSOUT=A
//SYSOUT    DD  SYSOUT=A
//SYSUDUMP  DD  SYSOUT=A
//TABLES    DD  DISP=SHR,DSNAME=HIS.TABLES
//MEMOIN    DD  DISP=&DISP,DSNAME=HIS.MEMOS
//MEMOOUT   DD  DISP=&DISP,DSNAME=HIS.MEMOS
//MEMOWK01  DD  UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//MEMOWK02  DD  UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//MEMOWK03  DD  UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//SORTLIB   DD  DISP=SHR,DSNAME=SYS1.SORTLIB
```

----- LISTPDS -----

```
//ACCIDENT  DD DISP=SHR,DSNAME=HIS.ACCIDENT
//ACCVEH    DD DISP=SHR,DSNAME=HIS.ACCVEH
```

LIBRARY=RUNLIB   MEMBER=HISRLG

```
//*     -----> HISRLG PROCEDURE <-----
//        PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//             OUTLIM=20000
//HIS EXEC PGM=HIS20400
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT   DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3    DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1    DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE      DD SYSOUT=Z,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT  DD SYSOUT=A
//PLIDUMP   DD SYSOUT=A
//SYSUDUMP  DD SYSOUT=A
//TABLES    DD DISP=SHR,DSNAME=HIS.TABLES
//ROADLOG   DD DISP=&DISP,DSNAME=HIS.ROADLOG
//TRUMILE   DD DISP=SHR,DSNAME=HIS.TRUMILE
//TRAFFIC   DD DISP=SHR,DSNAME=HIS.TRAFFIC
```

LIBRARY=RUNLIB   MEMBER=HISRRX

```
//*     -----> HISRRX PROCEDURE <-----
//        PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//             OUTLIM=20000
//HIS EXEC PGM=HIS20400
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT   DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER   DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3    DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1    DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE      DD SYSOUT=Z,OUTLIM=&OUTLIM,
//             DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
```

----- LISTPDS -----

```
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG
//TRUMILE  DD DISP=SHR,DSNAME=HIS.TRUMILE
//TRAFFIC DD DISP=SHR,DSNAME=HIS.TRAFFIC
//RAILROAD DD DISP=&DISP,DSNAME=HIS.RAILROAD
//RFREP    DD DISP=&DISP,DSNAME=HIS.RFREP
//SORTLIB  DD DISP=SHR,DSNAME=SYS1.SORTLIB
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
```

LIBRARY=RUNLIB   MEMBER=HISSKID

```
//*     -----> HISSKID PROCEDURE <-----
//      PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//           OUTLIM=20000
//HIS EXEC PGM=HIS20400
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTD
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//MAINTDIV DD DISP=SHR,DSNAME=HIS.MAINTDIV
//SKID     DD DISP=&DISP,DSNAME=HIS.SKID
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG,UNIT=(,,DEFER)
//TRUMILE  DD DISP=SHR,DSNAME=HIS.TRUMILE,UNIT=(,,DEFER)
```

----- LISTPDS -----

LIBRARY=FUNLIB   MEMBER=HISSUFF

```
//*    -----> HISSUFF PROCEDURE <-----
//       PROC FORM=A,FORM8=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//            OUTLIM=20000
//HIS  EXEC PGM=HIS20400
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//SORTLIB  DD DISP=SHR,DSNAME=SYS1.SORTLIB
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE8   DD SYSOUT=&FORM8,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CNJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG,UNIT=(,,DEFER)
//TRUMILE  DD DISP=SHR,DSNAME=HIS.TRUMILE
//TRAFFIC  DD DISP=SHR,DSNAME=HIS.TRAFFIC
//TRAFREP  DD DISP=SHR,DSNAME=HIS.TRAFREP
//ACCIDENT DD DISP=SHR,DSNAME=HIS.ACCIDENT
//ACCDIRI  DD DISP=SHR,DSNAME=HIS.ACCDIRI
//SUFFICY  DD DISP=&DISP,DSNAME=HIS.SUFFICY,UNIT=(,,DEFER)
//SUFFREP  DD DISP=&DISP,DSNAME=HIS.SUFFREP,UNIT=(,,DEFER)
//SORTIN   DD UNIT=SYSDA,SPACE=(CYL,3)
//SORTOUT  DD UNIT=SYSDA,DISP=OLD,VOL=REF=*.SORTIN,DSNAME=*.SORTIN
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(4),,CONTIG)
```

```
----- LISTPDS -----

        LIBRARY=RUNLIB          MEMBER=HISTRAF

//*     -----> HISTRAF PROCEDURE <-----
//      PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//           OUTLIM=20000
//HIS EXEC PGM=HIS2040
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//ROADLOG  DD DISP=SHR,DSNAME=HIS.ROADLOG,UNIT=(,,DEFER)
//TRUFILE  DD DISP=&DISP,DSNAME=HIS.TRUFILE
//TRAFFIC  DD DISP=&DISP,DSNAME=HIS.TRAFFIC
//TRAFREP  DD DISP=&DISP,DSNAME=HIS.TRAFREP


        LIBRARY=RUNLIB          MEMBER=HISJSN

//*     ----> HISJSN PROCEDURE <----
//      PROC FORM=A,COPIES=1,BLKSIZE=1330,RECFM=FBA,DISP=SHR,
//           SOR1=1,SOR2=1,OUTLIM=20000
//HIS EXEC PGM=HIS2040
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//INSTRCT  DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//PRINTER  DD SYSOUT=&FORM,COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//GREEN3   DD SYSOUT=(D,,0263),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//WHITE1   DD SYSOUT=(D,,0361),COPIES=&COPIES,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//CRJE     DD SYSOUT=Z,OUTLIM=&OUTLIM,
//            DCB=(BLKSIZE=&BLKSIZE,LRECL=133,RECFM=&RECFM)
//SYSPRINT DD SYSOUT=A
//PLIDUMP  DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//TABLES   DD DISP=SHR,DSNAME=HIS.TABLES
//SORTLIB  DD DISP=SHR,DSNAME=SYS1.SORTLIB
```

----- LISTPDS -----

```
//SORTIN    DD UNIT=SYSDA,SPACE=(CYL,(&SCR2,&SCR2))
//SORTOUT   DD DISP=(OLD,PASS),VOL=REF=*.SORTIN,DSNAME=*.SORTIN
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(&SCR1,&SCR1))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(&SCR1,&SCR1))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(&SCR1,&SCR1))
//SCRATCH1  DD DISP=(OLD,PASS),VOL=REF=*.SORTWK01,DSNAME=*.SORTWK01
//SCRATCH2  DD DISP=(OLD,PASS),VOL=REF=*.SORTWK02,DSNAME=*.SORTWK02
//SCRATCH3  DD DISP=(OLD,PASS),VOL=REF=*.SORTWK03,DSNAME=*.SORTWK03
//USN001    DD DISP=SHR,DSNAME=HIS.USN001,UNIT=(,,DEFER)
//USN001J   DD DISP=&DISP,DSNAME=HIS.USN001,UNIT=(,,DEFER)
//USN052    DD DISP=SHR,DSNAME=HIS.USN052,UNIT=(,,DEFER)
//USN052J   DD DISP=&DISP,DSNAME=HIS.USN052,UNIT=(,,DEFER)
//CTYX001   DD DISP=&DISP,DSNAME=HIS.CTYX001,UNIT=(,,DEFER)
//CTYX052   DD DISP=&DISP,DSNAME=HIS.CTYX052,UNIT=(,,DEFER)
//GRIDTBL   DD DISP=&DISP,DSNAME=HIS.GRIDTBL,UNIT=(,,DEFER)
//SGNXREF   DD DISP=&DISP,DSNAME=HIS.SGNXREF,UNIT=(,,DEFER)
//SIGNEDT   DD DISP=&DISP,DSNAME=HIS.SIGNEDT,UNIT=(,,DEFER)
```

LIBRARY=RUNLIB   MEMBER=HISPAN

```
//*   ---> HISPAN PROCEDURE <---
//       PROC FORMPAN=A,PANLIB='SYS1.PANLIB'
//PAN  EXEC PGM=HISUT006
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//PANDD1    DD DISP=SHR,DSNAME=&PANLIB
//SYSPRINT  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//OUT2      DD SYSOUT=Z
//OUT1      DD SYSOUT=&FORMPAN
```

LIBRARY=RUNLIB   MEMBER=HIS4VO

```
//*    ------ HIS4VO PROCEDURE ------
//       PROC CLIB='HIS.OBJECT',C=NCT,F=NOFLOW,G=NGS,
//            FORM=A,
//            L=LIST,M=MAP,O=OBJ,OPT='OPT(0)',COPIES=1
//PAN  EXEC PGM=PAN#1
//PANDD1    DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//             SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT  DD SYSOUT=&FORM,DCB=(RECFM=FB,LRECL=121,BLKSIZE=1331)
//PLI  EXEC PGM=IEL0AA,
//            PARM='AG,A,ESD,NEST,CF,STG,X,&C,&F,&G,&L,&M,&O,&OPT'
//SYSPRINT  DD SYSOUT=&FORM,COPIES=&COPIES,
```

----- LISTPDS -----

```
//                DCB=(BLKSIZE=1254,LRECL=125,RECFM=VBA)
//SYSUT1     DD UNIT=SYSDA,SPACE=(1024,(60,60),,CONTIG),DCB=BLKSIZE=1024
//SYSLIN     DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKOBJ(&DATASET),
//                SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSIN      DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB   DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=&FORM
//INLIB      DD DISP=(OLD,DELETE),DSNAME=*.PLI.SYSLIN
//OUTLIB     DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN      DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

LIBRARY=RUNLIB   MEMBER=HIS4VOL

```
//*    ------ HIS4VOL PROCEDURE ------
//      PROC OLIB='HIS.OBJECT',C=NCT,F=NOFLOW,G=NGS,
//             LET=,NCAL=,SIZE=,OVLY=,XREF=MAP,LCOND=7,
//             FORM=A,
//             LLIB='HIS.LOADTST',LKEDBLK=7294,
//             L=LIST,M=MAP,O=OBJ,OPT='OPT(0)',COPIES=1
//PAN EXEC PGM=PAN#1
//PANDD1     DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2     DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//                SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FB,LRECL=121,BLKSIZE=1331)
//PLI EXEC PGM=IELOAA,
//             PARM='AG,A,ESD,NEST,OF,STG,X,&C,&F,&G,&L,&M,&O,&OPT'
//SYSPRINT DD SYSOUT=&FORM,COPIES=&COPIES,
//                DCB=(BLKSIZE=1254,LRECL=125,RECFM=VBA)
//SYSUT1     DD UNIT=SYSDA,SPACE=(1024,(60,60),,CONTIG),DCB=BLKSIZE=1024
//SYSLIN     DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKOBJ(&DATASET),
//                SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSIN      DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB   DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=&FORM
//INLIB      DD DISP=(OLD,DELETE),DSNAME=*.PLI.SYSLIN
//OUTLIB     DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN      DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
//LKED   EXEC   PGM=IEWL,PARM='LIST,&XREF,&LET,&NCAL,&SIZE,&OV...
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=6...)
//SYSLIB   DD DSN=SYS1.PLILIB,DISP=SHR
//             DD DSN=HIS.SUBRTN4,DISP=SHR
//SYSUT1     DD UNIT=SYSSQ,SPACE=(1024,(200,20))
//SYSLMOD  DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKLOD(&DATASET),
//                DCB=BLKSIZE=&LKEDBLK,
//                SPACE=(CYL,(1,1,1))
```

----- LISTPDS -----

```
//LOADTST  DD DISP=SHR,DSNAME=HIS.LOADTST
//LOADLIB  DD DISP=SHR,DSNAME=HIS.LOADLIB
//OBJECT   DISP=SHR,DSNAME=HIS.OBJECT
//LOADSUB  DD DISP=SHR,DSNAME=HIS.LOADSUB
//REL4PTO  DD DISP=SHR,DSNAME=HIS.REL4PTO
//SUBRTN4  DD DISP=SHR,DSNAME=HIS.SUBRTN4
//SYSLIN DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//            DD DDNAME=SYSIN
//COPYLOD EXEC PGM=HISCOPY
//STEPLIB  DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.LKED.SYSLMOD
//OUTLIB    DD DISP=SHR,DSNAME=&LLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

LIBRARY=RUNLIB   MEMBER=HISCVD

```
//*    ------ HISCVD PROCEDURE ------
//      PROC OLIB='HIS.OBJECT',C=NCT,F=NOFLOW,G=NGS,
//            FORM=A,
//            L=LIST,M=MAP,O=OBJ,OPT='OPT(0)'
//PAN   EXEC  PGM=HISUT006
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//PANDD1    DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//                 SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(2,1))
//OUT2      DD SYSOUT=Z
//OUT1      DD SYSOUT=&FORM
//PLI EXEC PGM=IEL0AA,
//            PARM='AG,A,ESD,NEST,OF,STG,X,&C,&F,&G,&L,&M,&O,&OPT'
//SYSPRINT DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(3,1)),
//                 DCB=(BLKSIZE=1254,LRECL=125,RECFM=VBA)
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(60,60),,CONTIG),DCB=BLKSIZE=10
//SYSLIN    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKOBJ(&DATASET),
//                 SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB
//SYSIN     DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.PLI.SYSLIN
//OUTLIB    DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
//PLIEDIT EXEC PGM=HISUT001
//STEPLIB DD DISP=SHR,DSNAME=HIS.LOADTST
//           DD DISP=SHR,DSNAME=HIS.REL4PTO
//INPUT DD UNIT=SYSDA,VOL=REF=*.PLI.SYSPRINT,DISP=(OLD,PASS),
```

```
//             DSNAME=*.PLI.SYSPRINT
//OUT1   DD SYSOUT=&FORM
//OUT2   DD SYSOUT=Z
```

LIBRARY=RUNLIB   MEMBER=HIS4VP

```
//*    ------ HIS4VP PROCEDURE ------
//        PROC   OLIB='HIS.OBJECT',
//               OBJECT=LIST,STMT=NOSTMT,OPT=0,LOAD=LOAD
//PAN    EXEC   PGM=PAN#1
//PANDD1    DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//             SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FB,LRECL=121,BLKSIZE=1331)
//PL1L   EXEC   PGM=IEMAA,
//             PARM='NT,A,X,E,SM=(2,72,1),&OBJECT,&STMT,&LOAD,OPT=&OPT'
//SYSPRINT DD SYSOUT=A,DCB=(BLKSIZE=1254,LRECL=125,RECFM=VBA)
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(60,60),,CONTIG)
//SYSUT3    DD UNIT=SYSSQ,SPACE=(80,(250,250))
//SYSLIN    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKLOD(&DATASET),
//             SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSIN    DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB   DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=A
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.PL1L.SYSLIN
//OUTLIB    DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

LIBRARY=RUNLIB   MEMBER=HISCVA

```
//*    ------ HISCVA PROCEDURE ------
//       PROC   OLIB='HIS.OBJECT',FORM=A
//PAN   EXEC   PGM=HISUT006
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//PANDD1    DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//              SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(2,1))
//OUT2       DD SYSOUT=Z
//OUT1       DD SYSOUT=&FORM
//ASM   EXEC   PGM=IEUASM,PARM=DECK
//SYSPRINT DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(2,1)),
```

----- LISTPDS -----

```
//        DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3388)
//SYSLIB    DD DSNAME=SYS1.MACLIB,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(30,5))
//SYSUT2    DD UNIT=SYSDA,SPACE=(TRK,(30,5))
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,(80,5))
//SYSPUNCH DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKOBJ(&DATASET),
//             SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB
//SYSIN    DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB  DD DISP=SHR,DSNAME=HIS.REL4PTO
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.ASM.SYSPUNCH
//OUTLIB    DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
//ASMEDIT EXEC PGM=HISUT000
//STEPLIB DD DISP=SHR,DSN=HIS.LOADTST
//         DD DISP=SHR,DSN=HIS.REL4PTO
//INPUT    DD UNIT=SYSDA,VOL=REF=*.ASM.SYSPRINT,DISP=(OLD,PASS),
//            DSNAME=*.ASM.SYSPRINT
//OUT1      DD SYSOUT=&FORM
//OUT2     DD SYSOUT=Z
```

LIBRARY=RUNLIB   MEMBER=HIS4VA

```
//*    ------ HIS4VA PROCEDURE ------
//       PROC  OLIB='HIS.OBJECT',FORM=A
//PAN   EXEC  PGM=PAN#1
//PANDD1   DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2   DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//            SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FB,LRECL=121,BLKSIZE=1331)
//ASM   EXEC  PGM=IEUASM,PARM=DECK
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3388)
//SYSLIB    DD DSNAME=SYS1.MACLIB,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(30,5))
//SYSUT2    DD UNIT=SYSDA,SPACE=(TRK,(30,5))
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,(80,5))
//SYSPUNCH DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKOBJ(&DATASET),
//            SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSIN    DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB  DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.ASM.SYSPUNCH
//OUTLIB    DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

----- LISTPDS -----

```
        LIBRARY=RUNLIB          MEMBER=HIS4VAL

//*    ------ HIS4VAL PROCEDURE ------
//         PROC   OLIB='HIS.OBJECT',LLIB='HIS.LOADTST',
//            LKEDBLK=7294,FORM=A,
//            LCOND=8,LET=,NCAL=,OVLY=,SIZE=,XREF=MAP
//PAN     EXEC   PGM=PAN#1
//PANDD1    DD DISP=SHR,DSNAME=SYS1.PANLIB
//PANDD2    DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKSET,
//             SPACE=(CYL,(1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FB,LRECL=121,BLKSIZE=1331)
//ASM     EXEC   PGM=IEUASM,PARM=DECK
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3388)
//SYSLIB    DD DSNAME=SYS1.MACLIB,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(30,5))
//SYSUT2    DD UNIT=SYSDA,SPACE=(TRK,(30,5))
//SYSUT3    DD UNIT=SYSDA,SPACE=(TRK,(80,5))
//SYSPUNCH DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKOBJ(&DATASET),
//             SPACE=(CYL,(1,1,1)),DCB=(BLKSIZE=400,LRECL=80,RECFM=FB)
//SYSIN    DD DISP=(OLD,DELETE),DSNAME=*.PAN.PANDD2
//COPYOBJ EXEC PGM=HISCOPY
//STEPLIB   DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.ASM.SYSPUNCH
//OUTLIB    DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
//LKED    EXEC   PGM=IEWL,PARM='LIST,&XREF,&LET,&NCAL,&SIZE,&OVLY',
//             COND=(&LCOND,LT,ASM)
//SYSPRINT DD SYSOUT=&FORM,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=605)
//SYSLIB    DD DSNAME=SYS1.LINKLIB,DISP=SHR
//          DD DSNAME=HIS.SUBRTN4,DISP=SHR
//SYSUT1    DD UNIT=SYSSQ,SPACE=(1024,(200,20))
//SYSLMOD   DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKLOD(&DATASET),
//             DCB=BLKSIZE=&LKEDBLK,
//             SPACE=(CYL,(1,1,1))
//LOADTST   DD DISP=SHR,DSNAME=HIS.LOADTST
//LOADLIB   DD DISP=SHR,DSNAME=HIS.LOADLIB
//REL4PTO   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SUBRTN4   DD DISP=SHR,DSNAME=HIS.SUBRTN4
//OBJECT    DD DISP=SHR,DSNAME=HIS.OBJECT
//LOADSUB   DD DISP=SHR,DSNAME=HIS.LOADSUB
//SYSLIN    DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//          DD DDNAME=SYSIN
//COPYLOD EXEC PGM=HISCOPY
//STEPLIB   DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.LKED.SYSLMOD
//OUTLIB    DD DISP=SHR,DSNAME=&LLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

---- LISTPDS -----

```
LIBRARY=RUNLIB          MEMBER=HIS4L

//*     ------ HIS4L PROCEDURE ------
//       PROC   LET=,NCAL=,SIZE=,OVLY=,XREF=MAP,
//              FORM=A,
//              LKEDBLK=7294,PLILIB='SYS1.PLILIB',
//           OLIB='HIS.OBJECT',LLIB='HIS.LOADTST'
//LKED   EXEC   PGM=HISUT004,PARM='LIST,&XREF,&LET,&NCAL,&SIZE,&OVLY'
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SYSPRINT  DD UNIT=SYSDA,SPACE=(TRK,(2,1)),
//             DCB=(BLKSIZE=1210,LRECL=121,RECFM=FB)
//OUT2      DD SYSOUT=Z
//OUT1      DD SYSOUT=&FORM
//SYSLIB    DD DISP=SHR,DSNAME=&PLILIB
//          DD DSN=HIS.SUBRTN4,DISP=SHR
//SYSUT1    DD UNIT=SYSSQ,SPACE=(1024,(200,20))
//SYSLMOD   DD DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKLOD(&DATASET),
//             DCB=BLKSIZE=&LKEDBLK,
//             SPACE=(CYL,(1,1,1))
//LOADTST   DD DISP=SHR,DSNAME=HIS.LOADTST
//OBJECT    DD DISP=SHR,DSNAME=HIS.OBJECT
//REL4PTO   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SUBRTN4   DD DISP=SHR,DSNAME=HIS.SUBRTN4
//SYSLIN DD DISP=SHR,DSNAME=&OLIB.(&DATASET)
//          DD DDNAME=SYSIN
//COPYLOD EXEC PGM=HISCOPY
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SYSPRINT DD SYSOUT=&FORM
//INLIB     DD DISP=(OLD,DELETE),DSNAME=*.LKED.SYSLMOD
//OUTLIB    DD DISP=SHR,DSNAME=&LLIB.(&DATASET)
//SYSIN     DD DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

```
LIBRARY=RUNLIB          MEMBER=HIS4LREP

//*     ------ HIS4LREP PROCEDURE ------
//         PROC   LET=,NCAL=,SIZE=,OVLY=,XREF=MAP,
//                FORM=A,
//                PLILIB='SYS1.PLILIB',
//                LLIB='HIS.LOADTST',LKEDBLK=7294
//LKED   EXEC   PGM=HISUT004,
//                PARM='LIST,&XREF,&LET,&NCAL,&SIZE,&OVLY'
//STEPLIB   DD DISP=SHR,DSNAME=HIS.REL4PTO
//SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(2,1)),
//                DCB=(BLKSIZE=1210,LRECL=121,RECFM=FB)
//OUT2      DD SYSOUT=Z
//OUT1      DD SYSOUT=&FORM
//SYSLIB    DD DISP=SHR,DSNAME=&PLILIB
```

----- LISTPDS -----

```
//              DD  DSN=HIS.SUBRTN4,DISP=SHR
//SYSUT1        DD  UNIT=SYSDA,SPACE=(1024,(200,20))
//SYSLMOD       DD  DISP=(NEW,PASS),UNIT=SYSDA,DSNAME=&&WORKLOD(&DATASET),
//                      DCB=BLKSIZE=&LKEDBLK,
//                      SPACE=(CYL,(1,1,1))
//OBJECT        DD  DISP=SHR,DSNAME=HIS.OBJECT
//LOADTST       DD  DISP=SHR,DSNAME=HIS.LOADTST
//REL4PTO       DD  DISP=SHR,DSNAME=HIS.REL4PTO
//SUBRTN4       DD  DISP=SHR,DSNAME=HIS.SUBRTN4
//SYSLIN        DD  DDNAME=SYSIN
//COPYLOD EXEC PGM=HISCOPY
//STEPLIB       DD  DISP=SHR,DSNAME=HIS.REL4PTO
//SYSPRINT DD  SYSOUT=&FORM
//INLIB         DD  DISP=(OLD,DELETE),DSNAME=*.LKED.SYSLMOD
//OUTLIB        DD  DISP=SHR,DSNAME=&LLIB.(&DATASET)
//SYSIN         DD  DISP=SHR,DSNAME=SYS1.SORTPARM(HISCOPY)
```

LIBRARY=RUNLIB   MEMBER=HISSORTA

```
//*    ------ HISSORTA PROCEDURE ------
//SORT   EXEC  PGM=IEFRC000,PARM='MSG=AP'
//SYSOUT    DD  SYSOUT=A
//SORTLIB   DD  DSNAME=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(8),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(8),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(8),,CONTIG)
```

LIBRARY=RUNLIB   MEMBER=HISX133

```
//*    ------ HISX133 PROCEDURE ------
//   PROC   FORM=A
//PRINT EXEC PGM=PRNT133
//STEPLIB DD DISP=SHR,DSNAME=HIS.LOADLIB
//SYSPRINT DD SYSOUT=A
//PRINTER DD SYSOUT=&FORM,DCB=(BLKSIZE=1330,LRECL=133,RECFM=FBA),
//             SPACE=(CYL,(3,3))
```

APPENDIX B

DATA ELEMENT NAMES USED IN SELECT STATEMENTS

Accident Detail File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
2	Decimal	DAC.#-FATALITIES	DAC-15
2	Decimal	DAC.#-INJURIES	DAC-16
2	Decimal	DAC.#-PEDESTRIANS	DAC-14
2	Decimal	DAC.#-VEHICLES	DAC-13
2	Character	DAC.ACCIDENT-MONTH	DAC-1
12	Character	DAC.ACCIDENT-NUMBER	DAC-1
2	Character	DAC.ACCIDENT-YEAR	DAC-1
3	Character	DAC.AGENCY	DAC-1
2	Decimal	DAC.ANALYSIS-1	DAC-26
2	Decimal	DAC.ANALYSIS-2	DAC-26
3	Character	DAC.BADGE-NUMBER	DAC-1
3/18	Dec/Char	DAC.CITY	DAC-3
1	Decimal	DAC.CLASS-OF-TRAFWAY	DAC-1∅
1	Decimal	DAC.COLLISION-TYPE	DAC-27
2/15	Dec/Char	DAC.COUNTY	DAC-4
1	Decimal	DAC.DAMAGE-SEVERITY	DAC-9
6	Date	DAC.DATE-ARRIVED	DAC-2
6	Date	DAC.DATE-NOTIFIED	DAC-2
6	Date	DAC.DATE-OCCURRED	DAC-2
2	Decimal	DAC.DAY-ARRIVED	DAC-2
2	Decimal	DAC.DAY-NOTIFIED	DAC-2
2	Decimal	DAC.DAY-OCCURRED	DAC-2
5	Character	DAC.DISTANCE	DAC-5
1	Character	DAC.ENG-STUDY	DAC-25
2	Decimal	DAC.FIRST-EVENT	DAC-6
2	Decimal	DAC.FIRST-OBJECT	DAC-7
2	Decimal	DAC.HOUR-ARRIVED	DAC-2
2	Decimal	DAC.HOUR-NOTIFIED	DAC-2
2	Decimal	DAC.HOUR-OCCURRED	DAC-2
1	Decimal	DAC.INJURY-SEVERITY	DAC-8
1	Character	DAC.INVESTIGATED	DAC-29
1	Decimal	DAC.JUNCTION-LOCATION	DAC-12
1	Decimal	DAC.LIGHT-COND	DAC-19
12	Character	DAC.LOCATION	DAC-5
8	Character	DAC.MILEPOINT	DAC-5
2	Decimal	DAC.MINUTE-ARRIVED	DAC-2
2	Decimal	DAC.MINUTE-NOTIFIED	DAC-2
2	Decimal	DAC.MINUTE-OCCURRED	DAC-2
2	Decimal	DAC.MONTH-ARRIVED	DAC-2
2	Decimal	DAC.MONTH-NOTIFIED	DAC-2
2	Decimal	DAC.MONTH-OCCURRED	DAC-2

Length	Format	Data Element Name	Described Under
1	Decimal	DAC.OTHER-DAM-OWNER	DAC-23
1	Decimal	DAC.OTHER-DAM-SEV	DAC-22
2	Decimal	DAC.OTHER-DAM-TYPE	DAC-22
1	Character	DAC.PLUS	DAC-5
2	Decimal	DAC.POSTED-SPEED	DAC-24
3	Character	DAC.RANGE	DAC-5
8	Character	DAC.RANGE-TOWN-&-SECTN	DAC-5
3	Character	DAC.REFERENCE-POST	DAC-5
1	Character	DAC.REPORTABLE	DAC-28
1	Decimal	DAC.ROAD-COND	DAC-18
1	Decimal	DAC.ROADWAY-LOCATION	DAC-11
3	Character	DAC.ROUTE-NUMBER	DAC-5
4	Character	DAC.ROUTE-SYS-&-NUMBER	DAC-5
1	Character	DAC.ROUTE-SYSTEM	DAC-5
2	Character	DAC.SECTION	DAC-5
2	Character	DAC.SEQUENCE-NUMBER	DAC-1
3	Character	DAC.TOWNSHIP	DAC-5
2	Decimal	DAC.TRAF-CONTROLS	DAC-2Ø
1	Decimal	DAC.WEATHER-COND	DAC-17
4	Character	DAC.X-COORDINATE	DAC-5
4	Character	DAC.Y-COORDINATE	DAC-5
2	Decimal	DAC.YEAR-ARRIVED	DAC-2
2	Decimal	DAC.YEAR-NOTIFIED	DAC-2
2	Decimal	DAC.YEAR-OCCURRED	DAC-2

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
12	Character	DAC.ACCIDENT-NUMBER	DAC-1
2	Character	DAC.ACCIDENT-YEAR	DAC-1
3	Character	DAC.AGENCY	DAC-1
3	Character	DAC.BADGE-NUMBER	DAC-1
2	Character	DAC.ACCIDENT-MONTH	DAC-1
2	Character	DAC.SEQUENCE-NUMBER	DAC-1
6	Date	DAC.DATE-OCCURRED	DAC-2
2	Decimal	DAC.MONTH-OCCURRED	DAC-2
2	Decimal	DAC.DAY-OCCURRED	DAC-2
2	Decimal	DAC.YEAR-OCCURRED	DAC-2
2	Decimal	DAC.HOUR-OCCURRED	DAC-2
2	Decimal	DAC.MINUTE-OCCURRED	DAC-2
6	Date	DAC.DATE-NOTIFIED	DAC-2
2	Decimal	DAC.MONTH-NOTIFIED	DAC-2
2	Decimal	DAC.DAY-NOTIFIED	DAC-2
2	Decimal	DAC.YEAR-NOTIFIED	DAC-2
2	Decimal	DAC.HOUR-NOTIFIED	DAC-2
2	Decimal	DAC.MINUTE-NOTIFIED	DAC-2

Length	Format	Data Element Name	Described Under
6	Date	DAC.DATE-ARRIVED	DAC-2
2	Decimal	DAC.MONTH-ARRIVED	DAC-2
2	Decimal	DAC.DAY-ARRIVED	DAC-2
2	Decimal	DAC.YEAR-ARRIVED	DAC-2
2	Decimal	DAC.HOUR-ARRIVED	DAC-2
2	Decimal	DAC.MINUTE-ARRIVED	DAC-2
3/18	Dec/Char	DAC.CITY	DAC-3
2/15	Dec/Char	DAC.COUNTY	DAC-4
12	Character	DAC.LOCATION	DAC-5
1	Character	DAC.ROUTE-SYSTEM	DAC-5
3	Character	DAC.ROUTE-NUMBER	DAC-5
4	Character	DAC.ROUTE-SYS-&-NUMBER	DAC-5
3	Character	DAC.REFERENCE-POST	DAC-5
1	Character	DAC.PLUS	DAC-5
5	Character	DAC.DISTANCE	DAC-5
8	Character	DAC.MILEPOINT	DAC-5
4	Character	DAC.X-COORDINATE	DAC-5
4	Character	DAC.Y-COORDINATE	DAC-5
3	Character	DAC.RANGE	DAC-5
3	Character	DAC.TOWNSHIP	DAC-5
2	Character	DAC.SECTION	DAC-5
8	Character	DAC.RANGE-TOWN-&-SECTN	DAC-5
2	Decimal	DAC.FIRST-EVENT	DAC-6
2	Decimal	DAC.FIRST-OBJECT	DAC-7
1	Decimal	DAC.INJURY-SEVERITY	DAC-8
1	Decimal	DAC.DAMAGE-SEVERITY	DAC-9
1	Decimal	DAC.CLASS-OF-TRAFWAY	DAC-1Ø
1	Decimal	DAC.ROADWAY-LOCATION	DAC-11
1	Decimal	DAC.JUNCTION-LOCATION	DAC-12
2	Decimal	DAC.#-VEHICLES	DAC-13
2	Decimal	DAC.#-PEDESTRIANS	DAC-14
2	Decimal	DAC.#-FATALITIES	DAC-15
2	Decimal	DAC.#-INJURIES	DAC-16
1	Decimal	DAC.WEATHER-COND	DAC-17
1	Decimal	DAC.ROAD-COND	DAC-18
1	Decimal	DAC.LIGHT-COND	DAC-19
2	Decimal	DAC.TRAF-CONTROLS	DAC-2Ø
2	Decimal	DAC.OTHER-DAM-TYPE	DAC-21
1	Decimal	DAC.OTHER-DAM-SEV	DAC-22
1	Decimal	DAC.OTHER-DAM-OWNER	DAC-23
2	Decimal	DAC.POSTED-SPEED	DAC-24
1	Character	DAC.ENG-STUDY	DAC-25
2	Decimal	DAC.ANALYSIS-1	DAC-26
2	Decimal	DAC.ANALYSIS-2	DAC-26
1	Decimal	DAC.COLLISION-TYPE	DAC-27
1	Character	DAC.REPORTABLE	DAC-28
1	Character	DAC.INVESTIGATED	DAC-29

DAC-1 - Accident Number - The accident number is a 12-character field. It has two formats - one for investigated accidents and another for uninvestigated accidents. The investigated format is:

Columns	Length	Contents
1-2	2	Year occurred
3-5	3	Investigating agency
6-8	3	Badge number of investigating officer
9-1Ø	2	Month occurred
11-12	2	Sequence number

The uninvestigated format is:

Columns	Length	Contents
1-2	2	Year occurred
3	1	'R' if rural or 'M' if municipal
4-6	3	County number if rural or city number if municipal
7-8	2	Month occurred
9-12	4	Sequence number

Investigating agency codes include:

'ØØØ'	State highway patrol
'nnn'	(ØØ1-126) City number of city police agency
'Cnn'	(CØ1-C56) County number of county sheriff agency

City and county numbers are listed in tables 2-1 and 2-2 of chapter 2.

Data element names implemented for the accident number are:

Data Element Name	Columns	Length	Accident Number Format
DAC.ACCIDENT-NUMBER	1-12	12	Either
DAC.ACCIDENT-YEAR	1-2	2	Either
DAC.AGENCY	3-5	3	Investigated only
DAC.BADGE-NUMBER	6-8	3	Investigated only
DAC.ACCIDENT-MONTH	9-1Ø	2	Investigated only
DAC.SEQUENCE-NUMBER	11-12	2	Investigated only

When using any of the names that are meaningful only for investigated accidents, you should include the data element DAC.INVESTIGATED to select only the investigated accidents (see DAC-29).

Examples:

```
DAC.ACCIDENT-NUMBER *EQ* '74ØØØ356Ø2Ø2'

DAC.ACCIDENT-YEAR *EQ* '73'

DAC.INVESTIGATED *EQ* ' '  *AND*  DAC.AGENCY *EQ* 'Ø52'

DAC.INVESTIGATED *EQ* ' '  *AND*  DAC.BADGE-NUMBER *EQ* '355'

DAC.INVESTIGATED *EQ* ' '  *AND*  DAC.ACCIDENT-MONTH *EQ* '1Ø'

DAC.INVESTIGATED *EQ* ' '  *AND*  DAC.SEQUENCE-NUMBER *EQ* 'Ø1'
```

DAC-2 - Dates and Times - Each accident record contains space for storing the date and time occurred, the date and time the investigating agency was notified, and the date and time the investigating officer arrived at the accident scene. The date occurred is always stored. The time occurred may be stored but may be omitted (zeroes are stored). Midnight is stored as an hour of 24 and a minute of ØØ. Times between 12:Ø1 and 12:59 A.M. are stored as an hour of ØØ and the minute in the minute field. Dates and times notified and arrived may or may not be present for investigated accidents but are never present for uninvestigated accidents. Data element names used for dates and times are:

Data Element Name	Example
DAC.DATE-OCCURRED	DAC.DATE-OCCURRED *GT* 4/1/75
DAC.DATE-NOTIFIED	DAC.DATE-NOTIFIED *LE* 6/1/76
DAC.DATE-ARRIVED	DAC.DATE-ARRIVED *EQ* 12/25/75
DAC.MONTH-OCCURRED	DAC.MONTH-OCCURRED *EQ* 6
DAC.MONTH-NOTIFIED	(same format)
DAC.MONTH-ARRIVED	(same format)
DAC.DAY-OCCURRED	DAC.DAY-OCCURRED *GT* 25
DAC.DAY-NOTIFIED	(same format)
DAC.DAY-ARRIVED	(same format)
DAC.YEAR-OCCURRED	DAC.YEAR-OCCURRED *EQ* 75
DAC.YEAR-NOTIFIED	(same format)
DAC.YEAR-ARRIVED	(same format)
DAC.HOUR-OCCURRED	DAC.HOUR-OCCURRED *EQ* 2
DAC.HOUR-NOTIFIED	(same format)
DAC.HOUR-ARRIVED	(same format)
DAC.MINUTE-OCCURRED	DAC.MINUTE-OCCURRED *GT* 4Ø
DAC.MINUTE-NOTIFIED	(same format)
DAC.MINUTE-ARRIVED	(same format)

DAC-3 - City Number or City Name - The data element name is DAC.CITY. The comparison can be made by city number (DAC.CITY *EQ* 52) or by city name (DAC.CITY *EQ* 'GREAT-FALLS'). City names and numbers are shown in table 2-1 in chapter 2.

DAC-4 - County Number or County Name - The data element name is DAC.COUNTY. The comparison can be made by county number (DAC.COUNTY *EQ* 2) or by county name (DAC.COUNTY *EQ* 'CASCADE'). If using county number, use the registration numbering system. County names and numbers are shown in table 2-2 in chapter 2.

DAC-5 - Location Field - The location field can have any of several different formats:

Format 1 - No location coded: The entire field contains blanks or zeroes.

Format 2 - Route number only:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L,1-9)
2-4	3	Route number
5-12	8	8 blanks or 8 zeroes

Format 3 - Route number and milepoint:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L,1-9)
2-4	3	Route number
5-12	8	Milepoint in format nnn+nnnn or nnn-nnnn

Format 4 - Route number and coordinates:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L,1-9)
2-4	3	Route number
5-8	4	X-coordinate
9-12	4	Y-coordinate

Format 5 - Coordinates only:

Columns	Length	Contents
1	1	'M'
2-4	3	3 blanks, 3 zeroes, or city number
5-8	4	X-coordinate
9-12	4	Y-coordinate

Format 6 - Range Township and Section:

Columns	Length	Contents
1	1	'R'
2	1	Blank or zero
3-4	2	X- and Y-coordinate within section
5-7	3	Range (nnE or nnW)
8-1Ø	3	Township (nnN or nnS)
11-12	2	Section (nn)

Format 7 - Uninterpretable: Any format other than the above.

Data element names that can be used to refer to the location field are:

Data Element Name	Columns	Length
DAC.LOCATION	1-12	12
DAC.ROUTE-SYSTEM	1	1
DAC.ROUTE-NUMBER	2-4	3
DAC.ROUTE-SYS-&-NUMBER	1-4	4

Data Element Name	Columns	Length
DAC.REFERENCE-POST	5-7	3
DAC.PLUS	8	1
DAC.DISTANCE	9-12	4
DAC.MILEPOINT	5-12	8
DAC.X-COORDINATE	5-8	4
DAC.Y-COORDINATE	9-12	4
DAC.RANGE	5-7	3
DAC.TOWNSHIP	8-1Ø	3
DAC.SECTION	11-12	2
DAC.RANGE-TOWN-&-SECTN	5-12	8

Examples:

DAC.LOCATION *EQ* ' ' *OR* DAC.LOCATION *EQ* 'ØØØØØØØØØØØØ'

DAC.ROUTE-SYSTEM *EQ* 'P' *AND* DAC.ROUTE-NUMBER *EQ* 'ØØ8'

DAC.ROUTE-SYS-&-NUMBER *EQ* 'PØØ8'

DAC.REFERENCE-POST *GE* '1ØØ' *AND* DAC.REFERENCE-POST *LE* '15Ø'
AND DAC.PLUS *EQ* '+'

DAC.MILEPOINT *GE* '1ØØ+Ø2ØØ' *AND* DAC.PLUS *EQ* '+'

<u>DAC-6 - First Harmful Event</u> - The data element name is DAC.FIRST-EVENT. First harmful event codes are:

Ø Not stated
1 Overturned
2 Other non-collision
3 Collision with pedestrian
4 Collision with motor vehicle in transport
5 Collision with motor vehicle in other roadway
6 Collision with parked motor vehicle
7 Collision with railway train
8 Collision with pedalcycle
9 Collision with animal
1Ø Collision with fixed object
11 Collision with other object

Example: DAC.FIRST-EVENT *EQ* 1Ø

<u>DAC-7 - First Object Hit off Roadway</u> - The data element name is DAC.FIRST-OBJECT. First object codes are:

ØØ No object hit (may be coded as 24)
Ø1 End of overpass or river crossing
Ø2 Guardrail protecting overpass structure
Ø3 Overpass railing or side of overpass
Ø4 End of underpass
Ø5 Pier of underpass
Ø6 Guardrail protecting underpass

Ø7 Lighting pole, power pole, or signal pole
Ø8 Guardrail protecting lighting pole or power pole
Ø9 Sign
1Ø Guardrail protecting sign
11 Median guardrail
12 Guardrail along fill
13 End of guardrail
14 Other guardrail
15 Tree
16 Cut slope
17 Road approach
18 Rock or boulder
19 End of drainage pipe
2Ø Building or other structure
21 Fence
22 Raised median or curb
23 Other object
24 No object (may be coded as ØØ)
25 Unknown

Example: DAC.FIRST-OBJECT *EQ* 19

DAC-8 - Injury Severity - The data element name is DAC.INJURY-SEVERITY. The injury severity codes are:

Ø No injury
1 Fatal injury
2 Incapacitating injury
3 Non-incapacitating injury
4 Possible injury

(this field indicates the most severe injury that occurred in the accident)

Example: DAC.INJURY-SEVERITY *EQ* 1

DAC-9 - Damage Severity - The data element name is DAC.DAMAGE-SEVERITY. The damage severity codes are:

Ø No damage
1 Disabling damage
2 Functional damage
3 Other motor vehicle damage

(this field indicates the most severe damage that occurred in the accident)

Example: DAC.DAMAGE-SEVERITY *EQ* 1

DAC-1Ø - Class of Trafficway - The data element name is DAC.CLASS-OF-TRAFWAY. The class of trafficway codes are:

1 Interstate
3 Other US numbered route
4 Other state numbered route
6 County road
7 Local street
8 Other road

Example: DAC.CLASS-OF-TRAFWAY *LE* 4

DAC-11 - Roadway-Related Location - The data element name is DAC.ROADWAY-LOCATION. The roadway-related location codes are:

1 On roadway
2 Off roadway

Example: DAC.ROADWAY-LOCATION *EQ* 2

DAC-12 - Junction-Related Location - The data element name is DAC.JUNCTION-LOCATION. The junction-related location codes are:

Ø Non-junction
1 Intersection
2 Intersection-related
3 Driveway access

Example: DAC.JUNCTION-LOCATION *EQ* 1

DAC-13 - Number of Vehicles - The data element name is DAC.#-VEHICLES. The data item contains a count of the number of vehicles involved in the accident and always ranges from 1 to 99.

Example: DAC.#-VEHICLES *GE* 3

DAC-14 - Number of Pedestrians - The data element name is DAC.#-PEDESTRIANS. The data item contains a count of the number of pedestrians involved in the accident and always ranges from Ø to 99.

Example: DAC.#-PEDESTRIANS *NE* Ø

DAC-15 - Number of Fatalities - The data element name is DAC.#-FATALITIES. The data item contains a count of the number of fatalities in the accident, and always ranges from Ø to 99.

Example: DAC.#-FATALITIES *GE* 4

DAC-16 - Number of Injuries - The data element name is DAC.#-INJURIES. The data item contains a count of the number of injuries (other than fatalities) in the accident, and always ranges from Ø to 99.

Example: DAC.#-INJURIES *GE* 5

DAC-17 - Weather Condition - The data element name is DAC.WEATHER-COND. The weather condition codes are:

Ø	Not stated	3	Snowing
1	Clear	4	Fog
2	Raining	5	Other

Example: DAC.WEATHER-COND *EQ* 3

DAC-18 - Road Condition - The data element name is DAC.ROAD-COND. The road condition codes are:

Ø	Not stated	3	Snowy
1	Dry	4	Icy
2	Wet	5	Other

Example: DAC.ROAD-COND *EQ* 4

DAC-19 - Light Condition - The data element name is DAC.LIGHT-COND. The light condition codes are:

Ø	Not stated	3	Darkness, lighted
1	Daylight	4	Darkness, unlighted
2	Dawn or dusk	5	Other

Example: DAC.LIGHT-COND *EQ* 4

DAC-2Ø - Traffic Controls - The data element name is DAC.TRAF-CONTROLS. The traffic controls codes are:

ØØ No traffic control devices
Ø1 Traffic signals
Ø2 Traffic signals not working
Ø3 Traffic signals with pedestrian heads
Ø4 Traffic signals with pedestrian heads - heads not working
Ø5 Flasher

Ø6	Flasher not working	12	Railroad gates not working
Ø7	Stop sign	13	Do not enter signs
Ø8	Yield sign	14	Other regulatory sign
Ø9	Railroad signals	15	Warning sign
1Ø	Railroad signals not working	16	Pavement markings
11	Railroad gates		

Example: DAC.TRAF-CONTROLS *EQ* 5

DAC-21 - Other Damage Type - The data element name is DAC.OTHER-DAM-TYPE. The other damage type codes are:

ØØ No other damage
Ø1 Signal pole, lighting pole, or power pole
Ø2 Sign

Ø3	Guardrail	Ø9	Road surface
Ø4	Bridge	1Ø	Drainage structure
Ø5	Building	11	Fence
Ø6	Shrubbery or trees	12	Barricades
Ø7	Maintenance equipment	13	Other
Ø8	Fire hydrant		

Example: DAC.OTHER-DAM-TYPE *NE* Ø

DAC-22 - Other Damage Severity - The data element name is DAC.OTHER-DAM-SEV. The damage severity codes are:

Ø	No other damage	2	Moderate damage
1	Minor damage	3	Major damage

Example: DAC.OTHER-DAM-SEV *EQ* 3

DAC-23 - Other Damage Owner - The data element name is DAC.OTHER-DAM-OWNER. The other damage owner codes are:

Ø	No other damage	3	County
1	Federal	4	City
2	State	5	Private

Example: DAC.OTHER-DAM-OWNER *EQ* 5

DAC-24 - Posted Speed - The data element name is DAC.POSTED-SPEED. The data item contains the posted speed at the accident location.

Example: DAC.POSTED-SPEED *EQ* 55

DAC-25 - Engineering Study Requested - The data element name is DAC.ENG-STUDY. Code DAC.ENG-STUDY *EQ* 'X' to select accidents in which a study was requested and DAC.ENG-STUDY *NE* 'X' to select accidents in which no study was requested.

DAC-26 - Analysis (Contributing Circumstances) - The two analysis fields are referenced by the names DAC.ANALYSIS-1 and DAC.ANALYSIS-2. The analysis codes are:

ØØ No additional contributing circumstances
Ø1 Failed to have vehicle under control (speed not involved)
Ø2 Inattentive driving
Ø3 Inexperience
Ø4 Blackout, heart attack, stroke, etc.
Ø5 Fell asleep
Ø6 Sun glare
Ø7 Raining
Ø8 Snowing
Ø9 Whiteout
1Ø Blowing snow
11 Whiteout - meeting or following vehicle
12 Dust storm
13 Dust caused by wind or preceding vehicle on unoiled road surface
14 Road slippery or icy
15 Other weather conditions
16 Improper hitch
17 Blow out - flat tire
18 Stone thrown by vehicle
19 Avoiding another vehicle
2Ø Avoiding pedestrian - unexpected actions
21 Striking or avoiding domestic animal in roadway
22 Striking or avoiding wild animal in roadway
23 Striking or avoiding object in roadway

24 Distraction within vehicle
25 Distraction outside vehicle
26 Unwarranted slowing
27 Blinded by glaring lights other than vehicle
28 Passenger fell from vehicle
29 Occupant releases vehicle
3∅ Indian in violation on reservation - Patrol has no jurisdiction
31 Traffic control sign missing, down, etc.
32 Wind blowing
33 Water on highway
34 Fog
35 Load shifted
36 Bleeding pavement

Example: DAC.ANALYSIS-1 *EQ* 1∅ *OR* DAC.ANALYSIS-2 *EQ* 1∅

DAC-27 - Collision Type - The data element name is DAC.COLLISION-TYPE. The collision type codes are:

1 Head on
2 Rear end
3 Angle
4 Sideswipe meeting
5 Sideswipe passing
6 Backed into
7 Other

Example: DAC.COLLISION-TYPE *EQ* 1

DAC-28 - Reportable - The data element name is DAC.REPORTABLE. Code as DAC.REPORTABLE *EQ* 'X' to select legally reportable accidents and as DAC.REPORTABLE *NE* 'X' to select unreportable accidents.

DAC-29 - Investigated - The data element name is DAC.INVESTIGATED. Code as DAC.INVESTIGATED *EQ* ' ' to select investigated accidents and as DAC.INVESTIGATED *NE* ' ' to select uninvestigated accidents.

Accident Vehicle File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
2	Character	VAC.ACCIDENT-MONTH	VAC-1
12	Character	VAC.ACCIDENT-NUMBER	VAC-1
2	Character	VAC.ACCIDENT-YEAR	VAC-1
2	Decimal	VAC.AGE-1 - 6	VAC-11
3	Character	VAC.AGENCY	VAC-1
1	Decimal	VAC.ALCOHOL-1 - 6	VAC-8
3	Character	VAC.BADGE-NUMBER	VAC-1
6	Character	VAC.BIRTH-DATE	VAC-4
2	Decimal	VAC.BODY-STYLE	VAC-14
6	Character	VAC.CHARGE-CODE	VAC-6

Length	Format	Data Element Name	Described Under
1	Character	VAC.DAMAGE-LEVEL	VAC-17
1	Decimal	VAC.DAMAGE-SEVERITY	VAC-18
2	Character	VAC.DRIVER-STATE	VAC-3
1	Decimal	VAC.INJ-SEV-1 - 6	VAC-1Ø
2	Decimal	VAC.INTENT	VAC-13
1	Character	VAC.INTERSTATE-TRAF	VAC-16
1	Decimal	VAC.MECH-DEFECTS	VAC-7
1	Decimal	VAC.PHYS-DEFECTS	VAC-7
1	Decimal	VAC.POSSIBLE-VIOL	VAC-7
1	Character	VAC.RE-EXAM	VAC-5
2	Character	VAC.RECORD-NUMBER	VAC-2
1	Character	VAC.RECORD-TYPE	VAC-2
3	Character	VAC.RECORD-TYPE-&-NUMBER	VAC-2
1	Decimal	VAC.ROAD-DEFECTS	VAC-7
2	Character	VAC.SEQUENCE-NUMBER	VAC-1
1	Character	VAC.SEX-1 - 6	VAC-9
1	Character	VAC.TRAILER-STYLE	VAC-15
2	Decimal	VAC.VEH-YEAR	VAC-12
1	Decimal	VAC.VISION	VAC-7

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
12	Character	VAC.ACCIDENT-NUMBER	VAC-1
2	Character	VAC.ACCIDENT-YEAR	VAC-1
3	Character	VAC.AGENCY	VAC-1
3	Character	VAC.BADGE-NUMBER	VAC-1
2	Character	VAC.ACCIDENT-MONTH	VAC-1
2	Character	VAC.SEQUENCE-NUMBER	VAC-1
3	Character	VAC.RECORD-TYPE-&-NUMBER	VAC-2
1	Character	VAC.RECORD-TYPE	VAC-2
2	Character	VAC.RECORD-NUMBER	VAC-2
2	Character	VAC.DRIVER-STATE	VAC-3
6	Character	VAC.BIRTH-DATE	VAC-4
1	Character	VAC.RE-EXAM	VAC-5
6	Character	VAC.CHARGE-CODE	VAC-6
1	Decimal	VAC.VISION	VAC-7
1	Decimal	VAC.PHYS-DEFECTS	VAC-7
1	Decimal	VAC.ROAD-DEFECTS	VAC-7
1	Decimal	VAC.MECH-DEFECTS	VAC-7
1	Decimal	VAC.POSSIBLE-VIOL	VAC-7
1	Decimal	VAC.ALCOHOL-1 - 6	VAC-8
1	Character	VAC.SEX-1 - 6	VAC-9
1	Decimal	VAC.INJ-SEV-1 - 6	VAC-1Ø
2	Decimal	VAC.AGE-1 - 6	VAC-11
2	Decimal	VAC.VEH-YEAR	VAC-12
2	Decimal	VAC.INTENT	VAC-13
2	Decimal	VAC.BODY-STYLE	VAC-14
1	Decimal	VAC.TRAILER-STYLE	VAC-15
1	Character	VAC.INTERSTATE-TRAF	VAC-16
1	Character	VAC.DAMAGE-LEVEL	VAC-17
1	Decimal	VAC.DAMAGE-SEVERITY	VAC-18

VAC-1 - Accident Number - The data element names and format of the accident number are described above in DAC-1.

VAC-2 - Record Type and Number - These three characters in conjunction with the accident number provide a unique identifier for each record in the vehicle file. The first character is a type code:

'A' Vehicle record
'B' Pedestrian record
'C' Additional injury record

The remaining two characters contain a sequence number. The data element names for this field are:

Data Element Name	Columns	Length
VAC.RECORD-TYPE-&-NUMBER	1-3	3
VAC.RECORD-TYPE	1	1
VAC.RECORD-NUMBER	2-3	2

Example: VAC.RECORD-TYPE *EQ* 'A'

VAC-3 - Driver License State - The data element name is VAC.DRIVER-STATE. The data item is a 2-character state code. The state codes are the standard post office codes (eg., 'MT' for Montana). The codes are too numerous to list here but can be found in the Data Coding manual.

Example: VAC.DRIVER-STATE *EQ* 'MT'

VAC-4 - Date of Birth - The data element name is VAC.BIRTH-DATE. The data item is a 6-character field consisting of 2 digits each for month, day, and year. The entire field is left blank when the date of birth is unknown.

Example: VAC.BIRTH-DATE *EQ* 'Ø92748'

VAC-5 - Re-Examination Code - The data element name is VAC.RE-EXAM. Code as VAC.RE-EXAM *EQ* 'X' to select accidents in which a driver was recommended for re-examination or as VAC.RE-EXAM *NE* 'X' to select accidents in which no driver was recommended for re-examination.

VAC-6 - Charge Code - The data element name is VAC.CHARGE-CODE. The data item contains blanks if no summons was issued and a Montana Highway Patrol charge code if a summons was issued.

Example: VAC.CHARGE-CODE *NE* ' '

VAC-7 - Contributing Circumstances - The contributing circumstances are broken down into five categories: vision defects, physical defects, road defects, mechanical defects, and possible violations. The data element names are VAC. VISION, VAC.PHYS-DEFECTS, VAC.ROAD-DEFECTS, VAC.MECH-DEFECTS, and VAC.POSSIBLE-VIOL.

The vision codes are:

Ø Vision not obscured
1 Buildings
2 Trees or hedges
3 Other vehicle
4 Smoke
5 Dust
6 Other

The physical defects codes are:

Ø No apparent defects
1 Vision
2 Hearing
3 Illness
4 Missing Limbs
5 Other

The road defects codes are:

Ø No road defects
1 Holes or ruts
2 Shoulder
3 Loose material
4 Construction
5 Other

The mechanical defects codes are:

Ø No apparent defects
1 Lights
2 Brakes
3 Tires or steering
4 Other

The possible violation codes are:

Ø No apparent violations
1 Had been drinking
2 Reckless driving
3 Speed to fast for conditions
4 Failed to yield right-of-way
5 Improper passing
6 Improper backing
7 Improper turn
8 Fail to signal
9 Other

Examples:

VAC.VISION *EQ* 4

VAC.POSSIBLE-VIOL *GE* 5

VAC-8 - Alcohol - Six spaces are available in each record for indicating alcohol. The data element names are VAC.ALCOHOL-1, VAC.ALCOHOL-2,...,VAC.ALCOHOL-6. VAC.ALCOHOL-1 refers to the driver or pedestrian. The alcohol codes are:

Ø Had not been drinking
1 Had been drinking

Example: VAC.ALCOHOL-1 *EQ* 1

VAC-9 - Sex - Six spaces are available in each record for indicating sex. The data element names are VAC.SEX-1, VAC,SEX-2,...,VAC.SEX-6. VAC.SEX-1 refers to the driver or pedestrian. The sex codes are:

'M' Male
'F' Female
' ' Unknown

Example: VAC.SEX-1 *EQ* 'M'

VAC-1Ø - Injury Severity - Six spaces are available in the file for storing injury severity. The data element names are VAC.INJ-SEV-1, VAC.INJ-SEV-2,..., VAC.INJ-SEV-6. VAC.INJ-SEV-1 refers to the driver or pedestrian. The injury severity codes are:

Ø No injury
1 Fatal injury
2 Incapacitating injury (cannot perform normally)
3 Non-incapacitating injury (evidence of injury)
4 Possible injury (apparent symptoms)

Example: VAC.INJ-SEV-1 *EQ* 1

VAC-11 - Age - Six spaces are available in the file for storing age. The data element names are VAC.AGE-1, VAC.AGE-2,..., VAC.AGE-6. VAC.AGE-1 refers to the driver or pedestrian. The values stored range from Ø to 99. Ages older than 99 are stored as 99. Ages under 1 year are stored as 1. Unknown ages are stored as Ø.

Example: VAC.AGE-1 *GE* 15 *AND* VAC.AGE-1 *LE* 18

VAC-12 - Vehicle Year - The data element name is VAC.VEH-YEAR.

Example: VAC.VEH-YEAR *EQ* 7Ø

VAC-13 - Intent - The data element name is VAC.INTENT. The actual codes depend upon whether the record is a vehicle record or a pedestrian record. Vehicle intent codes are:

ØØ	Not stated	Ø6	Slow or stop
Ø1	Go straight ahead	Ø7	Start in traffic lane
Ø2	Overtake	Ø8	Start from parked position
Ø3	Make right turn	Ø9	Back
Ø4	Make left turn	1Ø	Remain stopped in traffic lane
Ø5	Make U turn	11	Remain parked

Pedestrian intent codes are:

ØØ Not stated
Ø1 Crossing at intersection or in crosswalk
Ø2 Crossing other than at intersection or in crosswalk
Ø3 Walking in roadway with traffic
Ø4 Walking in roadway against traffic
Ø5 Standing in roadway
Ø6 Pushing or working on vehicle in roadway
Ø7 Other working in roadway
Ø8 Playing in roadway
Ø9 Other in roadway
1Ø Not in roadway
11 Not stated (same as ØØ)

Example: VAC.RECORD-TYPE *EQ* 'A' *AND* VAC.INTENT *EQ* 9

VAC-14 - Body Style - The data element name is VAC.BODY-STYLE. The body style codes are:

ØØ	Not stated	Ø8	Motor cycle
Ø1	Passenger car	Ø9	Ambulance
Ø2	Mini bus or van	1Ø	Farm tractor or machinery
Ø3	Bus	11	Construction machinery
Ø4	School bus	12	Pickup with camper
Ø5	Pickup	13	Bicycle
Ø6	Truck or truck tractor	14	Snowmobile
Ø7	Motor home	15	Other

Example: VAC.BODY-STYLE *EQ* 8

VAC-15 - Trailer Style - The data element name is VAC.TRAILER-STYLE. The trailer style codes are:

Ø	No trailer	4	Boat trailer
1	Camping trailer	5	Semi trailer
2	Mobile home	6	Commercial cargo trailer
3	Utility trailer	7	Other trailer

Example: VAC.TRAILER-STYLE *EQ* 2

VAC-16 - Interstate Traffic - The data element name is VAC.INTERSTATE-TRAF. Use VAC.INTERSTATE-TRAF *EQ* 'X' to locate vehicles involved in interstate traffic and VAC.INTERSTATE-TRAF *NE* 'X' to locate other vehicles.

VAC-17 - Damage Level - The data element name is VAC.DAMAGE-LEVEL. Use VAC.DAMAGE-LEVEL *EQ* 'X' to locate vehicles whose damage exceeds $25Ø and VAC.DAMAGE-LEVEL *NE* 'X' to locate other vehicles.

VAC-18 - Damage Severity - The data element name is VAC.DAMAGE-SEVERITY. The damage severity codes are:

Ø	No damage	2	Function damage
1	Disabling damage	3	Other motor vehicle damage

Example: VAC.DAMAGE-SEVERITY *EQ* 1

Accident Directory File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
2	Decimal	ACD.#-FATALITIES	ACD-3
2	Decimal	ACD.#-INJURIES	ACD-4
1	Decimal	ACD.#-LANES	ACD-9
2	Character	ACD.ACCIDENT-MONTH	ACD-2
12	Character	ACD.ACCIDENT-NUMBER	ACD-2
2	Character	ACD.ACCIDENT-YEAR	ACD-2
3	Character	ACD.AGENCY	ACD-2
3	Character	ACD.BADGE-NUMBER	ACD-2
1	Decimal	ACD.COLLISION-TYPE	ACD-7
6	Date	ACD.DATE-OCCURRED	ACD-5
2	Decimal	ACD.DAY-OCCURRED	ACD-5
6	Character	ACD.DISTANCE	ACD-1
2	Decimal	ACD.FIRST-EVENT	ACD-6
2	Decimal	ACD.HOUR-OCCURRED	ACD-5
15	Character	ACD.KEY	ACD-1
9	Character	ACD.MILEPOINT	ACD-1
2	Decimal	ACD.MONTH-OCCURRED	ACD-5
3	Character	ACD.REFERENCE-POST	ACD-1
1	Decimal	ACD.ROAD-COND	ACD-8
5	Character	ACD.ROUTE-NUMBER	ACD-1
6	Character	ACD.ROUTE-SYS-&-NUMBER	ACD-1
1	Character	ACD.ROUTE-SYSTEM	ACD-1
2	Character	ACD.SEQUENCE-NUMBER	ACD-2
2	Decimal	ACD.YEAR-OCCURRED	ACD-5

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
15	Character	ACD.KEY	ACD-1
6	Character	ACD.ROUTE-SYS-&-NUMBER	ACD-1
1	Character	ACD.ROUTE-SYSTEM	ACD-1
5	Character	ACD.ROUTE-NUMBER	ACD-1
9	Character	ACD.MILEPOINT	ACD-1
3	Character	ACD.REFERENCE-POST	ACD-1
6	Character	ACD.DISTANCE	ACD-1
12	Character	ACD.ACCIDENT-NUMBER	ACD-2
2	Character	ACD.ACCIDENT-YEAR	ACD-2
3	Character	ACD.AGENCY	ACD-2
3	Character	ACD.BADGE-NUMBER	ACD-2
2	Character	ACD.ACCIDENT-MONTH	ACD-2
2	Character	ACD.SEQUENCE-NUMBER	ACD-2
2	Decimal	ACD.#-FATALITIES	ACD-3
2	Decimal	ACD.#-INJURIES	ACD-4

Length	Format	Data Element Name	Described Under
6	Date	ACD.DATE-OCCURRED	ACD-5
2	Decimal	ACD.MONTH-OCCURRED	ACD-5
2	Decimal	ACD.DAY-OCCURRED	ACD-5
2	Decimal	ACD.YEAR-OCCURRED	ACD-5
2	Decimal	ACD.HOUR-OCCURRED	ACD-5
2	Decimal	ACD.FIRST-EVENT	ACD-6
1	Decimal	ACD.COLLISION-TYPE	ACD-7
1	Decimal	ACD.ROAD-COND	ACD-8
1	Decimal	ACD.#-LANES	ACD-9

ACD-1 - Key - The key is stored in the following format:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L)
2-6	5	Route number
7-15	9	Milepoint in format 'nnn+n.nnn'

The following data element names are used for referencing the key:

Data Element Name	Columns	Length
ACD.KEY	1-15	15
ACD.ROUTE-SYS-&-NUMBER	1-6	6
ACD.ROUTE-SYSTEM	1	1
ACD.ROUTE-NUMBER	2-6	5
ACD.MILEPOINT	7-15	9
ACD.REFERENCE-POST	7-9	3
ACD.DISTANCE	1Ø-15	6

Examples:

ACD.ROUTE-SYS-&-NUMBER *EQ* 'PØØØØ8'

ACD.MILEPOINT *GE* 'Ø5Ø+Ø.588'

ACD-2 - Accident Number - The data element names and format of the accident number are described above in DAC-1.

ACD-3 - Number of Fatalities - See DAC-15 above.

ACD-4 - Number of Injuries - See DAC-16 above.

ACD-5 - Date and Time Occurred - See DAC-2 above (note than the minute is not stored in the directory file).

ACD-6 - First Harmful Event - See DAC-6 above.

ACD-7 - Collision Type - See DAC-27 above.

ACD-8 - Road Condition - See DAC-18 above.

ACD-9 - Number of Lanes - The data element name is ACD.#-LANES. The data items contains the number of lanes and always ranges from Ø (unknown) to 9.

Example: ACD.#-LANES *EQ* 4

Bridge File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
4	Decimal	BDG.#-APPR-SPANS	BDG-25
1	Decimal	BDG.#-LANES-ON	BDG-17
1	Decimal	BDG.#-LANES-UNDER	BDG-17
3	Decimal	BDG.#-MAIN-SPANS	BDG-25
1	Character	BDG.APPR-ALIGN-RATING	BDG-44
1	Character	BDG.APPR-ALIGNMENT	BDG-41
4	Character	BDG.APPR-GUARDRAIL	BDG-38
3	Decimal	BDG.APPR-STRUCT-TYPE	BDG-24
(4,1)	Decimal	BDG.BDG-DECK-WIDTH	BDG-3Ø
(4,1)	Decimal	BDG.BDG-ROADWAY-WIDTH	BDG-29
1	Decimal	BDG.BRIDGE-MEDIAN	BDG-19
1	Character	BDG.CHAN-&-CHAN-PROTECT	BDG-41
16	Character	BDG.COIN-KEY	BDG-3
9	Character	BDG.COIN-MILEPOINT	BDG-3
5	Character	BDG.COIN-ROUTE	BDG-3
6	Character	BDG.COIN-SYS-&-ROUTE	BDG-3
1	Character	BDG.COIN-SYSTEM	BDG-3
2	Decimal	BDG.CONSTRUCTION-DIST	BDG-5
5	Decimal	BDG.COST-OF-IMPROVEMENTS	BDG-56
1	Character	BDG.CULVERT-&-RET-WALLS	BDG-41
1	Decimal	BDG.CUSTODIAN	BDG-15
1	Character	BDG.DECK-COND	BDG-41
1	Character	BDG.DECK-GEOMETRY	BDG-44
6	Decimal	BDG.DESIGN-ADT	BDG-52
4	Decimal	BDG.DESIGN-LOAD	BDG-18
3	Decimal	BDG.DETOUR-LENGTH	BDG-6
6	Character	BDG.DISTANCE	BDG-1
15	Character	BDG.FACILITY-CARRIED	BDG-13
25	Character	BDG.FEATURES-INTERSECTED	BDG-7
4	Decimal	BDG.HORIZ-NAV-CLEAR	BDG-22
6	Decimal	BDG.IMPROVEMENT-LENGTH	BDG-48
6	Decimal	BDG.INSPECTION-DATE	BDG-57
3	Decimal	BDG.INVENTORY-RATING	BDG-43
9	Decimal	BDG.INVENTORY-ROUTE	BDG-12
16	Character	BDG.KEY	BDG-1
2	Decimal	BDG.LATITUDE-DEGREES	BDG-11
(3,1)	Decimal	BDG.LATITUDE-MINUTES	BDG-11
(3,1)	Decimal	BDG.LEFT-SIDEWALK-WIDTH	BDG-28
3	Decimal	BDG.LONGITUDE-DEGREES	BDG-11
(3,1)	Decimal	BDG.LONGITUDE-MINUTES	BDG-11

Length	Format	Data Element Name	Described Under
3	Decimal	BDG.MAIN-STRUCT-TYPE	BDG-24
2	Decimal	BDG.MAINTENANCE-DIV	BDG-4
1	Character	BDG.MAJOR-OR-MINOR	BDG-1Ø
4	Decimal	BDG.MAX-SPAN-LENGTH	BDG-26
6	Character	BDG.MICROFILM-SER-#	BDG-59
9	Character	BDG.MILEPOINT	BDG-1
(3,1)	Decimal	BDG.MIN-LAT-UNDER-LEFT	BDG-35
(3,1)	Decimal	BDG.MIN-LAT-UNDER-RIGHT	BDG-35
2	Decimal	BDG.MIN-VERT-CLEAR-FT	BDG-8
2	Decimal	BDG.MIN-VERT-CLEAR-IN	BDG-8
2	Decimal	BDG.MIN-VERT-OVER-FT	BDG-33
2	Decimal	BDG.MIN-VERT-OVER-IN	BDG-33
2	Decimal	BDG.MIN-VERT-UNDER-FT	BDG-34
2	Decimal	BDG.MIN-VERT-UNDER-IN	BDG-34
1	Decimal	BDG.NAV-CONTROL	BDG-22
3	Decimal	BDG.OPERATING-RATING	BDG-43
1	Decimal	BDG.PHYSICAL-VULNER	BDG-14
6	Character	BDG.POSTED-LOAD	BDG-4Ø
2	Decimal	BDG.POSTED-SPEED	BDG-39
15	Character	BDG.PROJECT-NUMBER	BDG-32
2	Decimal	BDG.PROP-#-LANES	BDG-51
1	Decimal	BDG.PROP-DESIGN-LOAD	BDG-49
4	Decimal	BDG.PROP-ROAD-WIDTH	BDG-5Ø
3	Character	BDG.REFERENCE-POST	BDG-1
2	Decimal	BDG.REMAINING-LIFE	BDG-42
1	Character	BDG.REMARK	BDG-2
(3,1)	Decimal	BDG.RIGHT-SIDEWALK-WIDTH	BDG-28
5	Character	BDG.ROUTE-NUMBER	BDG-1
6	Character	BDG.ROUTE-SYS-&-NUMBER	BDG-1
1	Character	BDG.ROUTE-SYSTEM	BDG-1
1	Character	BDG.SAFE-LOAD-CAPACITY	BDG-44
1	Character	BDG.SEQUENCE-NUMBER	BDG-1
2	Decimal	BDG.SERVICE-TYPE	BDG-23
2	Decimal	BDG.SKEW	BDG-2Ø
6	Decimal	BDG.STATION-NUMBER	BDG-31
6	Character	BDG.STRUCT-BATCH-#	BDG-58
6	Decimal	BDG.STRUCT-LENGTH	BDG-27
1	Character	BDG.STRUCTURAL-COND	BDG-44
1	Decimal	BDG.STRUCTURE-FLARE	BDG-21
1	Character	BDG.SUBSTRUCTURE	BDG-41
1	Character	BDG.SUPERSTRUCTURE	BDG-41
2	Decimal	BDG.SURFACE-DEPTH	BDG-37
(3,1)	Decimal	BDG.TOT-HORIZ-CLEAR	BDG-9
1	Decimal	BDG.TYPE-OF-SERVICE	BDG-46
3	Decimal	BDG.TYPE-OF-WORK	BDG-47
1	Decimal	BDG.TYPE-ROADWAY-IMPROVE	BDG-55
1	Character	BDG.UNDERCLEARANCES	BDG-44
6	Date	BDG.UPDATE-DATE	BDG-6Ø
2	Decimal	BDG.UPDATE-DAY	BDG-6Ø
2	Decimal	BDG.UPDATE-MONTH	BDG-6Ø
2	Decimal	BDG.UPDATE-YEAR	BDG-6Ø

Length	Format	Data Element Name	Described Under
3	Decimal	BDG.VERT-NAV-CLEAR	BDG-22
1	Character	BDG.WATERWAY-ADEQUACY	BDG-44
4	Decimal	BDG.WEARING-SURFACE	BDG-36
2	Decimal	BDG.YEAR-BUILT	BDG-16
2	Decimal	BDG.YEAR-IMPROVED	BDG-16
2	Decimal	BDG.YEAR-NEEDED-IMPROVE	BDG-45
2	Decimal	BDG.YEAR-OF-DESIGN-ADT	BDG-53
2	Decimal	BDG.YEAR-ROADWAY-IMPROVE	

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
16	Character	BDG.KEY	BDG-1
6	Character	BDG.ROUTE-SYS-&-NUMBER	BDG-1
1	Character	BDG.ROUTE-SYSTEM	BDG-1
5	Character	BDG.ROUTE-NUMBER	BDG-1
9	Character	BDG.MILEPOINT	BDG-1
3	Character	BDG.REFERENCE-POST	BDG-1
6	Character	BDG.DISTANCE	BDG-1
1	Character	BDG.SEQUENCE-NUMBER	BDG-1
1	Character	BDG.REMARK	BDG-2
16	Character	BDG.COIN-KEY	BDG-3
6	Character	BDG.COIN-SYS-&-ROUTE	BDG-3
1	Character	BDG.COIN-SYSTEM	BDG-3
5	Character	BDG.COIN-ROUTE	BDG-3
9	Character	BDG.COIN-MILEPOINT	BDG-3
2	Decimal	BDG.MAINTENANCE-DIV	BDG-4
2	Decimal	BDG.CONSTRUCTION-DIST	BDG-5
3	Decimal	BDG.DETOUR-LENGTH	BDG-6
25	Character	BDG.FEATURES-INTERSECTED	BDG-7
2	Decimal	BDG.MIN-VERT-CLEAR-FT	BDG-8
2	Decimal	BDG.MIN-VERT-CLEAR-IN	BDG-8
(3,1)	Decimal	BDG.TOT-HORIZ-CLEAR	BDG-9
1	Character	BDG.MAJOR-OR-MINOR	BDG-1Ø
2	Decimal	BDG.LATITUDE-DEGREES	BDG-11
(3,1)	Decimal	BDG.LATITUDE-MINUTES	BDG-11
3	Decimal	BDG.LONGITUDE-DEGREES	BDG-11
(3,1)	Decimal	BDG.LONGITUDE-MINUTES	BDG-11
9	Decimal	BDG.INVENTORY-ROUTE	BDG-12
15	Character	BDG.FACILITY-CARRIED	BDG-13
1	Decimal	BDG.PHYSICAL-VULNER	BDG-14
1	Decimal	BDG.CUSTODIAN	BDG-15
2	Decimal	BDG.YEAR-BUILT	BDG-16
2	Decimal	BDG.YEAR-IMPROVED	BDG-16
1	Decimal	BDG.#-LANES-ON	BDG-17
1	Decimal	BDG.#-LANES-UNDER	BDG-17
4	Decimal	BDG.DESIGN-LOAD	BDG-18
1	Decimal	BDG.BRIDGE-MEDIAN	BDG-19
2	Decimal	BDG.SKEW	BDG-2Ø
1	Decimal	BDG.STRUCTURE-FLARE	BDG-21

Length	Format	Data Element Name	Described Under
1	Decimal	BDG.NAV-CONTROL	BDG-22
3	Decimal	BDG.VERT-NAV-CLEAR	BDG-22
4	Decimal	BDG.HORIZ-NAV-CLEAR	BDG-22
2	Decimal	BDG.SERVICE-TYPE	BDG-23
3	Decimal	BDG.MAIN-STRUCT-TYPE	BDG-24
3	Decimal	BDG.APPR-STRUCT-TYPE	BDG-24
3	Decimal	BDG.#-MAIN-SPANS	BDG-25
4	Decimal	BDG.#-APPR-SPANS	BDG-25
4	Decimal	BDG.MAX-SPAN-LENGTH	BDG-26
6	Decimal	BDG.STRUCT-LENGTH	BDG-27
(3,1)	Decimal	BDG.LEFT-SIDEWALK-WIDTH	BDG-28
(3,1)	Decimal	BDG.RIGHT-SIDEWALK-WIDTH	BDG-28
(4,1)	Decimal	BDG.BDG-ROADWAY-WIDTH	BDG-29
(4,1)	Decimal	BDG.BDG-DECK-WIDTH	BDG-30
6	Decimal	BDG.STATION-NUMBER	BDG-31
15	Character	BDG.PROJECT-NUMBER	BDG-32
2	Decimal	BDG.MIN-VERT-OVER-FT	BDG-33
2	Decimal	BDG.MIN-VERT-OVER-IN	BDG-33
2	Decimal	BDG.MIN-VERT-UNDER-FT	BDG-34
2	Decimal	BDG.MIN-VERT-UNDER-IN	BDG-34
(3,1)	Decimal	BDG.MIN-LAT-UNDER-RIGHT	BDG-35
(3,1)	Decimal	BDG.MIN-LAT-UNDER-LEFT	BDG-35
4	Decimal	BDG.WEARING-SURFACE	BDG-36
2	Decimal	BDG.SURFACE-DEPTH	BDG-37
4	Character	BDG.APPR-GUARDRAIL	BDG-38
2	Decimal	BDG.POSTED-SPEED	BDG-39
6	Character	BDG.POSTED-LOAD	BDG-40
1	Character	BDG.DECK-COND	BDG-41
1	Character	BDG.SUPERSTRUCTURE	BDG-41
1	Character	BDG.SUBSTRUCTURE	BDG-41
1	Character	BDG.CHAN-&-CHAN-PROTECT	BDG-41
1	Character	BDG.CULVERT-&-RET-WALLS	BDG-41
1	Character	BDG.APPR-ALIGNMENT	BDG-41
2	Decimal	BDG.REMAINING-LIFE	BDG-42
3	Decimal	BDG.OPERATING-RATING	BDG-43
3	Decimal	BDG.INVENTORY-RATING	BDG-43
1	Character	BDG.STRUCTURAL-COND	BDG-44
1	Character	BDG.DECK-GEOMETRY	BDG-44
1	Character	BDG.UNDERCLEARANCES	BDG-44
1	Character	BDG.SAFE-LOAD-CAPACITY	BDG-44
1	Character	BDG.WATERWAY-ADEQUACY	BDG-44
1	Character	BDG.APPR-ALIGN-RATING	BDG-44
2	Decimal	BDG.YEAR-NEEDED-IMPROVE	BDG-45
1	Decimal	BDG.TYPE-OF-SERVICE	BDG-46
3	Decimal	BDG.TYPE-OF-WORK	BDG-47
6	Decimal	BDG.IMPROVEMENT-LENGTH	BDG-48
1	Decimal	BDG.PROP-DESIGN-LOAD	BDG-49
4	Decimal	BDG.PROP-ROAD-WIDTH	BDG-50
2	Decimal	BDG.PROP-#-LANES	BDG-51

Length	Format	Data Element Name	Described Under
6	Decimal	BDG.DESIGN-ADT	BDG-52
2	Decimal	BDG.YEAR-OF-DESIGN-ADT	BDG-53
2	Decimal	BDG.YEAR-ROADWAY-IMPROVE	BDG-54
1	Decimal	BDG.TYPE-ROADWAY-IMPROVE	BDG-55
5	Decimal	BDG.COST-OF-IMPROVEMENTS	BDG-56
6	Decimal	BDG.INSPECTION-DATE	BDG-57
6	Character	BDG.STRUCT-BATCH-#	BDG-58
6	Character	BDG.MICROFILM-SER-#	BDG-59
6	Date	BDG.UPDATE-DATE	BDG-6Ø
2	Decimal	BDG.UPDATE-MONTH	BDG-6Ø
2	Decimal	BDG.UPDATE-DAY	BDG-6Ø
2	Decimal	BDG.UPDATE-YEAR	

BDG-1 - Key - The key format of the bridge file is:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L,M,F,R)
2-6	5	Route number
7-15	9	Milepoint in format 'nnn+n.nnn'
16	1	Sequence number to allow multiple records at one location

The data element names that can be used with the key are:

Data Element Name	Columns	Length
BDG.KEY	1-16	16
BDG.ROUTE-SYS-&-NUMBER	1-6	6
BDG.ROUTE-SYSTEM	1	1
BDG.ROUTE-NUMBER	2-6	5
BDG.MILEPOINT	7-15	9
BDG.REFERENCE-POST	7-9	3
BDG.DISTANCE	1Ø-15	6
BDG.SEQUENCE-NUMBER	16	1

BDG-2 - Remark - The data element name is BDG.REMARK. Most records have a blank in the remark field but the following remark codes indicate special types of bridges:

'A' Adjacent opening of preceding structure
'C' Coincident route using same structure
'D' Dam
'F' River ford
'L' Opposite traffic lane of preceding structure
'P' Parallel or dual structure
'R' Structure serving section direction traffic only
'S' Structure serving opposing traffic only
'T' Temporary structure
'U' Underpass

Example: BDG.REMARK *EQ* ' '

BDG-3 - Coincident Key - The coincident key has the same format as the key (see BDG-1 above). The following data element names can be used to refer to the coincident key:

Data Element Name	Columns	Length
BDG.COIN-KEY	1-16	16
BDG.COIN-SYS-&-ROUTE	1-6	6
BDG.COIN-SYSTEM	1	1
BDG.COIN-ROUTE	2-6	5
BDG.COIN-MILEPOINT	7-15	9

BDG-4 - Maintenance Division - The data element name is BDG.MAINTENANCE-DIV. The maintenance divisions are:

11	21	31	41	51
12	22	32	42	52
				53

Example: BDG.MAINTENANCE-DIV *EQ* 32

BDG-5 - Construction District - The data element name is BDG.CONSTRUCTION-DIST. The 12 construction districts are numbered from Ø1 to 12.

Example: BDG.CONSTRUCTION-DIST *EQ* 7

BDG-6 - Detour Length - The data element name is BDG.DETOUR-LENGTH. The data item contains the detour length to the nearest mile. Special codes are:

ØØØ - Ground level bypass available
ØØ1 - Bridge is one of twin bridges - twin can be used as bypass

Example: BDG.DETOUR-LENGTH *LE* 1

BDG-7 - Features Intersected - The data element name is BDG.FEATURES-INTERSECTED. The data item contains a 25-character verbal description.

BDG-8 - Minimum Vertical Clearance - Data element names are BDG.MIN-VERT-CLEAR-FT (feet) and BDG.MIN-VERT-CLEAR-IN (inches). Both contain 99 when there are no vertical restrictions.

Example: BDG.MIN-VERT-CLEAR-FT *LT* 1Ø

BDG-9 - Total Horizontal Clearance - The data element name is BDG.TOT-HORIZ-CLEAR. The data item contains the horizontal clearance to the nearest foot and ranges from ØØ.Ø to 99.Ø.

Examples: BDG.TOT-HORIZ-CLEAR *LT* 1Ø
BDG.TOT-HORIZ-CLEAR *GT* 15.3

BDG-1Ø - Major or Minor - Use BDG.MAJOR-OR-MINOR *EQ* 'N' to locate minor bridges and BDG.MAJOR-OR-MINOR *NE* 'N' to locate major bridges (the data item contains a 'Y' for major bridges).

BDG-11 - Latitude and Longitude - Latitude is stored as nn degrees and nn.n minutes. Longitude is stored as nnn degrees and nn.n minutes. Data element names are:

Data Element Name
BDG.LATITUDE-DEGREES
BDG.LATITUDE-MINUTES
BDG.LONGITUDE-DEGREES
BDG.LONGITUDE-MINUTES

Example: BDG.LATITUDE-DEGREES *EQ* 46 *AND*

(BDG.LATITUDE-MINUTES *GE* 1Ø *AND* BDG.LATITUDE-MINUTES *LE* 1Ø.5)

BDG-12 - Inventory Route - The data element name is BDG.INVENTORY-ROUTE. The inventory route is a 9-digit field structured as:

Digits	Length	Contents
1	1	Carried by or goes under
2	1	Type of roadway
3	1	Special conditions
4-8	5	Signed route number
9	1	Direction

Codes used in the first digit are:

1 Route carried by structure
2 Route goes under structure

Type or roadway codes are:

1	Interstate	5	City street
2	U.S. highway	6	Federal lands road
3	State highway	7	State lands road
4	County highway	8	Other

Special condition codes are:

Ø	No special conditions	5	Toll roads
1	Mainline	6	Business
2	Alternate	7	Ramp or wye
3	Bypass	8	Service/frontage road
4	Spur	9	Truck route

Direction codes are:

Ø Not applicable

1	North	3	South
2	East	4	West

Example: BDG.INVENTORY-ROUTE *EQ* 12ØØØØ1Ø4

BDG-13 - Facility Carried - The data element name is BDG.FACILITY-CARRIED. The field contains a 15-character verbal description.

BDG-14 - Physical Vulnerability - The data element name is BDG.PHYSICAL-VULNER. The physical vulnerability codes are:

Ø	No structure	5	Suspension
1	Timber trestle	6	Reinforced concrete - massive arch
2	Concrete girder	7	Dam bridge
3	Steel girder	8	Box culverts
4	Cantilever and truss	9	Tunnels

Example: BDG.PHYSICAL-VULNER *EQ* 4

BDG-15 - Custodian - The data element name is BDG.CUSTODIAN. The custodian codes are:

1	State highway dept	6	Railroad
2	Other state agency	7	Other private agency
3	County agency	8	Combination
4	City/local agency	9	Unknown
5	Federal agency		

Example: BDG.CUSTODIAN *EQ* 5

BDG-16 - Year Built and Year Improved - Data element names are BDG.YEAR-BUILT and BDG.YEAR-IMPROVED.

Example: BDG.YEAR-BUILT *LE* 5Ø

BDG-17 - Number of Lanes - Data element names are BDG.#-LANES-ON (number of lanes on structure) and BDG.#-LANES-UNDER (number of lanes under structure). Both can range from Ø to 9.

Example: BDG.#-LANES-ON *GE* 4

BDG-18 - Design Load - The data element name is BDG.DESIGN-LOAD. The data item is a 4-digit number containing the HS rating (H in first two digits and S in last two digits). The following special codes are used:

ØØØØ Unknown bridge type
ØØØ7 Pedestrian bridge
ØØØ8 Railroad bridge
ØØØ9 Other bridge type

Example: BDG.DESIGN-LOAD *EQ* 8

BDG-19 - Bridge Median - The data element name is BDG.BRIDGE-MEDIAN. The bridge median codes are:

Ø No median
1 Open median
2 Closed median

Example: BDG.BRIDGE-MEDIAN *EQ* Ø

BDG-2Ø - Skew - The data element name is BDG.SKEW. The field contains the skew angle to the nearest degree. A value of 99 indicates a major variation in the skews of substructure units. Example: BDG.SKEW *EQ* 99.

BDG-21 - Structure Flared - The data element name is BDG.STRUCTURE-FLARE. The codes are:

Ø	Width of structure does not vary	Example:
1	Width of structure varies	BDG.STRUCTURE-FLARE *EQ* 1

BDG-22 - Navigation Control and Clearance - Data element names are:

BDG.NAV-CONTROL
BDG.VERT-NAV-CLEAR
BDG.HORIZ-NAV-CLEAR

Navigation control codes are:

Ø	No navigation control	Example:
1	Navigation control	BDG.NAV-CONTROL *EQ* 1

Clearances are specified to nearest foot and are non-zero only when navigation control is 1. Example: BDG.NAV-CONTROL *EQ* 1 *AND* BDG.VERT-NAV-CLEAR *LE* 1Ø

BDG-23 - Service Type - The data element name is BDG.SERVICE-TYPE. The field is a 2-digit field. The first field indicates service on the bridge:

1	Highway	6	Overpass or second level of multilevel intchg
2	Railroad	7	Third level of multilevel interchange
3	Pedestrian	8	Fourth level of multilevel interchange
4	Highway/railroad	9	Building or plaza
5	Highway/pedestrian	Ø	Other

The second digit indicates service under the bridge:

1	Highway	6	Highway/waterway
2	Railroad	7	Railroad/waterway
3	Pedestrian	8	Highway/railroad/waterway
4	Highway/railroad	9	Relief
5	Waterway	Ø	Other

Example: BDG.SERVICE-TYPE *GE* 5Ø *AND* BDG.SERVICE-TYPE *LE* 59.

BDG-24 - Structure Type - Data element names are BDG.MAIN-STRUCT-TYPE (main structure type) and BDG.APPR-STRUCT-TYPE (approach structure type). The first digit indicates design and material:

1	Concrete	6	Prestress concrete continuous
2	Concrete continuous	7	Timber
3	Steel	8	Masonry
4	Steel continuous	9	Aluminum, W.I., or C.I.
5	Prestress concrete	Ø	Other

The last two digits indicate type of design and/or construction:

Ø1	Slab	Ø5	Box beam/girders - multiple
Ø2	Stringer/multi-beam/girder	Ø6	Box beam/girders - single or spread
Ø3	Girder and floorbeam	Ø7	Frame
Ø4	Tee beam	Ø8	Orthotropic

Ø9	Truss - deck	15	Movable - lift
1Ø	Truss - thru	16	Movable - bascule
11	Arch - deck	17	Movable - swing
12	Arch - thru	18	Tunnel
13	Suspension	19	Culvert
14	Stayed girder	ØØ	Other

Example: BDG.APPR-STRUCT-TYPE *EQ* 218

BDG-25 - Number of Spans - Data element names are BDG.#-MAIN-SPANS (number of main spans) and BDG.#-APPR-SPANS (number of approach spans). The number of main spans varies from Ø to 999 and the number of approach spans varies from Ø to 9999. Example: BDG.#-MAIN-SPANS *GE* 5ØØ.

BDG-26 - Length of Longest Span - The data element name is BDG.MAX-SPAN-LENGTH. The field contains the length of the longest span to nearest foot and ranges from Ø to 9999. Example: BDG.MAX-SPAN-LENGTH *GE* 5ØØ.

BDG-27 - Structure Length - The data element name is BDG.STRUCTURE-LENGTH. The length is to nearest foot and ranges from Ø to 999999. Example: BDG.STRUCTURE-LENGTH *GE* 15ØØ.

BDG-28 - Sidewalk Width - The data element names are BDG.LEFT-SIDEWALK-WIDTH and BDG.RIGHT-SIDEWALK-WIDTH. Each width is in feet and ranges from ØØ.Ø to 99.9. Example: BDG.LEFT-SIDEWALK-WIDTH *GE* 5 *AND* BDG.LEFT-SIDEWALK-WIDTH *LE* 5.5.

BDG-29 - Roadway Width - The data element name is BDG.BDG-ROADWAY-WIDTH. The field contains the curb-to-curb width and varies from ØØØ.Ø to 999.9. Example: BDG.BDG-ROADWAY-WIDTH *GE* 1ØØ.

BDG-3Ø - Deck Width - The data element name is BDG.BDG-DECK-WIDTH. The field contains the out-to-out deck width and ranges from ØØØ.Ø to 999.9. Example: BDG.BDG-DECK-WIDTH *GE* 2ØØ.5.

BDG-31 - Station Number - The data element name is BDG.STATION-NUMBER. The field contains the station number (74Ø+ØØØ is stored as 74ØØØØ). Example: BDG.STATION-NUMBER *GE* 74ØØØØ.

BDG-32 - Project Number - The data element name is BDG.PROJECT-NUMBER. The field contains a 15-character project number.

BDG-33 - Minimum Vertical Overclearance - Feet and inches are stored separately in BDG.MIN-VERT-OVER-FT and BDG.MIN-VERT-OVER-IN. Both contain 99 when there are no restrictions. Example: BDG.MIN-VERT-OVER-FT *LT* 1Ø.

BDG-34 - Minimum Vertical Underclearance - Feet and inches are stored separately in BDG.MIN-VERT-UNDER-FT and BDG.MIN-VERT-UNDER-IN. Both contain zeroes if the structure is over anything other than a roadway or railway track. Example: BDG.MIN-VERT-UNDER-FT *GT* Ø *AND* BDG.MIN-VERT-UNDER-FT *LT* 2Ø.

BDG-35 - Minimum Lateral Underclearance - Data element names are BDG.MIN-LAT-UNDER-RIGHT and BDG.MIN-LAT-UNDER-LEFT. Each contains clearance in feet and ranges from ØØ.Ø to 99.9. 99.9 is stored when the structure is over anything other than a roadway or a railroad track. Example: BDG.MIN-LAT-UNDER-RIGHT *LE* 8.5.

BDG-36 - Wearing Surface - The data element name is BDG.WEARING-SURFACE. The field contains one of these codes:

ØØØ2	Open grate	4131	Alphalt
ØØØ3	Wood planking	7ØØ1	Portland cement concrete
2Ø1Ø	Gravel or stone	9999	Other

Example: BDG.WEARING-SURFACE *EQ* 7ØØ1.

BDG-37 - Wearing Surface Depth - The data element name is BDG.SURFACE-DEPTH. The field contains thickness in inches and ranges from Ø to 99. Example: BDG.SURFACE-DEPTH *LT* 5.

BDG-38 - Approach Guardrail - The data element name is BDG.APPR-GUARDRAIL. The field is a 4-character field. The first two characters indicate guard-rail type and the last two digits provide the height in inches.

BDG-39 - Posted Speed Limit - The data element is BDG.POSTED-SPEED. The field ranges from Ø (no posted speed limit) to 99. Example: BDG.POSTED-SPEED *EQ* 55.

BDG-4Ø - Posted Load Limit - The data element is BDG.POSTED-LOAD. The field contains blanks if there is no posted load limit and otherwise contains a load limit in tons that ranges from 'ØØØØØ1' to '999999'. Example: BDG.POSTED-LOAD *GE* 'ØØ1ØØØ'.

BDG-41 - Condition - Part 1 - These include the following data element names:

Data Element Name	Condition Of
BDG.DECK-COND	Deck
BDG.SUPERSTRUCTURE	Superstructure
BDG.SUBSTRUCTURE	Substructure
BDG.CHAN-&-CHAN-PROTECT	Channel and channel protection
BDG.CULVERT-&-RET-WALLS	Culvert and retaining walls
BDG.APPR-ALIGNMENT	Approach alignment

The following codes are used in each of these fields:

'N'	Not applicable	'4'	Minimum adequate to tolerate load
'9'	New condition	'3'	Warrants closing to trucks
'8'	Good condition	'2'	Warrants closing to all traffic
'7'	Minor items need repair	'1'	Repairable if desirable to reopen
'6'	Major items need repair	'Ø'	Condition beyond repair - may collapse
'5'	Major repairs needed		

Example: BDG.DECK-COND *NE* 'N' *AND* BDG.DECK-COND *LE* '4'

BDG-42 - Estimated Remaining Life - The data element name is BDG.REMAINING-LIFE. The field contains an estimate of the number of years of remaining life and ranges from Ø (not stated) to 99. Example: BDG.REMAINING-LIFE *GE* 5.

BDG-43 - Operating and Inventory Rating - Data element names are BDG.OPERATING-RATING and BDG.INVENTORY-RATING. Each contains a 3-digit value. Pedestrian loading is stored as 8ØØ. Railroad loading is stored using the Cooper class or equivalent. For other loading, the last two digits contain the gross loading in tons with one of these codes in the first digit:

1	H truck	6	3-3 trailer
2	HS truck	7	Railroad loading
3	Alternate interstate loading	8	Pedestrian/special loading
4	3-axle truck (type 3)	9	Gross load only given
5	3-S semi trailer		

Example: BDG.OPERATING-RATING *EQ* 8ØØ.

BDG-44 - Condition - Part 2 - These include the following data element names:

Data Element Name	Condition Of
BDG.STRUCTURAL-COND	Structure
BDG.DECK-GEOMETRY	Deck geometry
BDG.UNDERCLEARANCES	Underclearances
BDG.SAFE-LOAD-CAPACITY	Capacity
BDG.WATERWAY-ADEQUACY	Waterway
BDG.APPR-ALIGN-RATING	Approach alignment

The following codes are used in these fields:

'N'	Not applicable	'4'	Min. to leave in place as is
'9'	Superior to desirable	'3'	High repair priority
'8'	Equal to desirable	'2'	High replacement priority
'7'	Better than minimum	'1'	Immediate repair necessary
'6'	Equal to minimum	'Ø'	Immediate replacement necessary
'5'	Better than min. adequacy		

Example: BDG.DECK-GEOMETRY *NE* 'N' *AND* BDG.DECK-GEOMETRY *LE* '3'.

BDG-45 - Year of Needed Improvement - The data element name is BDG.YEAR-NEEDED-IMPROVE. Example: BDG.YEAR-NEEDED-IMPROVE *EQ* 78.

BDG-46 - Type of Service - The data element name is BDG.TYPE-OF-SERVICE. The field contains the proposed type of service using the codes shown above in BDG-23. Example: BDG.TYPE-OF-SERVICE *EQ* 1.

BDG-47 - Type of Work - The data element name is BDG.TYPE-OF-WORK. The field contains 3 digits. The last digit contains one of these codes:

1 Work to be done by contract
2 Work to be done by state forces

The first two digits contains one of the following codes:

3Ø	Widen existing structure
31	Replace because of condition
32	Replace due to road relocation
33	Construct new structure
34	Construct pedestrian structure
35	Other structure work
36	Strengthen
37	Rehabilitate

Example: BDG.TYPE-OF-WORK *EQ* 3Ø1

BDG-48 - Improvement Length - The data element name is BDG.IMPROVEMENT-LENGTH. The field contains the proposed improvement length to the nearest foot and ranges from Ø to 999999. Example: BDG.IMPROVEMENT-LENGTH *GE* 25ØØ.

BDG-49 - Proposed Design Loading - The data element name is BDG.PROP-DESIGN-LOAD. The codes are:

1	H 1Ø	6	HS 2Ø+
2	H 15	7	Pedestrian
3	HS 15	8	Railroad
4	H 2Ø	9	Other
5	HS 2Ø	Ø	Unknown

Example: BDG.PROP-DESIGN-LOAD *EQ* 3.

BDG-5Ø - Proposed Roadway Width - The data element name is BDG.PROP-ROAD-WIDTH. The field contains width in feet and ranges from Ø to 9999. Example: BDG.PROP-ROAD-WIDTH *GE* 1ØØØ.

BDG-51 - Proposed Number of Lanes - The data element name is BDG.PROP-#-LANES. The field ranges from Ø to 99. Example: BDG.PROP-#-LANES *GE* 4.

BDG-52 - Design ADT - The design ADT data element name is BDG.DESIGN-ADT. The ADT ranges from Ø to 999999. Example: BDG.DESIGN-ADT *GE* 15ØØ.

BDG-53 - Year of Design ADT - The data element name is BDG.YEAR-OF-DESIGN-ADT. Example: BDG.YEAR-OF-DESIGN-ADT *EQ* 78.

BDG-54 - Year of Roadway Improvements - The data element name is BDG.YEAR-ROADWAY-IMPROVE. Example: BDG.YEAR-ROADWAY-IMPROVE *EQ* 78.

BDG-55 - Type of Roadway Improvements - The data element name is BDG.TYPE-ROADWAY-IMPROVE. Codes are:

Ø	Not applicable	3	Widening
1	Resurface	4	Shoulder improvements
2	Reconstruction	5	Other

Example: BDG.TYPE-ROADWAY-IMPROVE *EQ* 3.

BDG-56 - Cost of Improvements - The data element name is BDG.COST-OF-IMPROVEMENTS. Cost is in thousands of dollars and ranges from Ø to 99999. Example: BDG.COST-OF-IMPROVEMENTS *GE* 5Ø.

BDG-57 - Inspection Date - The data element name is BDG.INSPECTION-DATE. The date is stored in six digits as month, day, year. Example: BDG.INSPECTION-DATE *EQ* Ø4Ø176.

BDG-58 - Structure Batch Serial Number - The data element name is BDG.STRUCT-BATCH-#. The field contains a 6-character batch number.

BDG-59 - Microfilm Serial Number - The data element name is BDG.MICROFILM-SER-#. The fields contains a 6-character serial number.

BDG-6Ø - Date of Update - The data element for the date of update field is BDG.UPDATE-DATE. Example: BDG.UPDATE-DATE *GE* 1/1/76. The individual components can be referenced by the names BDG.UPDATE-MONTH, BDG.UPDATE-DAY, and BDG.UPDATE-YEAR. Example: BDG.UPDATE-YEAR *GE* 76.

Railroad File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
1	Decimal	RRX.#-CROSSBUCKS	RRX-25
2	Decimal	RRX.#-DAILY-FREIGHT	RRX-14
1	Decimal	RRX.#-DAILY-PASSENGER	RRX-14
3	Decimal	RRX.#-DAILY-SWITCH	RRX-14
1	Decimal	RRX.#-FLASHING-LIGHTS	RRX-25
1	Decimal	RRX.#-GATES	RRX-25
1	Decimal	RRX.#-LANES	RRX-26
1	Decimal	RRX.#-MAIN-TRACKS	RRX-21
3	Decimal	RRX.#-NIGHTLY-TRAINS	RRX-27
2	Decimal	RRX.#-OTHER-TRACKS	RRX-21
2	Decimal	RRX.#-SEASONAL-FREIGHT	RRX-14
1	Decimal	RRX.#-SEASONAL-PASSENGER	RRX-14
3	Decimal	RRX.#-SEASONAL-SWITCH	RRX-14
1	Decimal	RRX.#-STOP-SIGNS	RRX-25
2	Decimal	RRX.#-WEEKLY-FREIGHT	RRX-14
1	Decimal	RRX.#-WEEKLY-PASSENGER	RRX-14
3	Decimal	RRX.#-WEEKLY-SWITCH	RRX-14
1	Decimal	RRX.#-WIGWAGS	RRX-25
3	Decimal	RRX.ANGLE-OR1	RRX-7
3	Decimal	RRX.ANGLE-OR2	RRX-7
15	Character	RRX.BRANCH-LINE	RRX-19
(6,2)	Decimal	RRX.BRANCH-LINE-MILEPOST	RRX-2Ø
7	Character	RRX.CROSSING-ID	RRX-18

Length	Format	Data Element Name	Described Under
2	Decimal	RRX.CURVATURE-DEG-1	RRX-1∅
2	Decimal	RRX.CURVATURE-DEG-2	RRX-1∅
2	Decimal	RRX.CURVATURE-MIN-1	RRX-1∅
2	Decimal	RRX.CURVATURE-MIN-2	RRX-1∅
2	Decimal	RRX.DAILY-FREIGHT-SPEED	RRX-15
2	Decimal	RRX.DAILY-PASS-SPEED	RRX-15
2	Decimal	RRX.DAILY-SWITCH-SPEED	RRX-15
6	Date	RRX.DATE-OF-UPDATE	RRX-29
2	Decimal	RRX.DAY-OF-UPDATE	RRX-29
1	Decimal	RRX.DIRECTION-1	RRX-6
1	Decimal	RRX.DIRECTION-2	RRX-6
6	Character	RRX.DISTANCE	RRX-1
3	Decimal	RRX.DISTANCE-1	RRX-9
3	Decimal	RRX.DISTANCE-2	RRX-9
4	Decimal	RRX.EST-ADT	RRX-28
2	Decimal	RRX.GRADE-1	RRX-11
2	Decimal	RRX.GRADE-2	RRX-11
6	Date	RRX.INVENTORY-DATE	RRX-16
2	Decimal	RRX.INVENTORY-DAY	RRX-16
2	Decimal	RRX.INVENTORY-MONTH	RRX-16
2	Decimal	RRX.INVENTORY-YEAR	RRX-16
15	Character	RRX.KEY	RRX-1
1	Decimal	RRX.LOCAL-INTERFERENCE-1	RRX-12
1	Decimal	RRX.LOCAL-INTERFERENCE-2	RRX-12
25	Character	RRX.LOCATION	RRX-4
9	Character	RRX.MILEPOINT	RRX-1
2	Decimal	RRX.MONTH-OF-UPDATE	RRX-29
4	Character	RRX.OPERATING-RAILROAD	RRX-17
4	Character	RRX.OTHER-RR-NAME	RRX-23
1	Decimal	RRX.OTHER-RR-USE	RRX-22
3	Character	RRX.REFERENCE-POST	RRX-1
2	Decimal	RRX.ROAD-WIDTH	RRX-5
5	Character	RRX.ROUTE-NUMBER	RRX-1
6	Character	RRX.ROUTE-SYS-&-NUMBER	RRX-1
1	Character	RRX.ROUTE-SYSTEM	RRX-1
2	Decimal	RRX.SEASONAL-FREIGHT-SPE	RRX-15
2	Decimal	RRX.SEASONAL-PASS-SPEED	RRX-15
2	Decimal	RRX.SEASONAL-SWITCH-SPEE	RRX-15
4	Decimal	RRX.SIGHT-DIST-LEFT1	RRX-8
4	Decimal	RRX.SIGHT-DIST-LEFT2	RRX-8
4	Decimal	RRX.SIGHT-DIST-RIGHT1	RRX-8
4	Decimal	RRX.SIGHT-DIST-RIGHT2	RRX-8
6	Date	RRX.SURVEY-DATE	RRX-2
2	Decimal	RRX.SURVEY-DAY	RRX-2
2	Decimal	RRX.SURVEY-MONTH	RRX-2
2	Decimal	RRX.SURVEY-YEAR	RRX-2
1	Character	RRX.URBAN-RURAL	RRX-3
1	Decimal	RRX.VERT-SIGHT-RESTR-1	RRX-13
1	Decimal	RRX.VERT-SIGHT-RESTR-2	RRX-13

Length	Format	Data Element Name	Described Under
2	Decimal	RRX.WEEKLY-FREIGHT-SPEED	RRX-15
2	Decimal	RRX.WEEKLY-PASS-SPEED	RRX-15
2	Decimal	RRX.WEEKLY-SWITCH-SPEED	RRX-15
2	Decimal	RRX.YEAR-OF-UPDATE	RRX-29

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
15	Character	RRX.KEY	RRX-1
6	Character	RRX.ROUTE-SYS-&-NUMBER	RRX-1
1	Character	RRX.ROUTE-SYSTEM	RRX-1
5	Character	RRX.ROUTE-NUMBER	RRX-1
9	Character	RRX.MILEPOINT	RRX-1
3	Character	RRX.REFERENCE-POST	RRX-1
6	Character	RRX.DISTANCE	RRX-1
6	Date	RRX.SURVEY-DATE	RRX-2
2	Decimal	RRX.SURVEY-MONTH	RRX-2
2	Decimal	RRX.SURVEY-DAY	RRX-2
2	Decimal	RRX.SURVEY-YEAR	RRX-2
1	Character	RRX.URBAN-RURAL	RRX-3
25	Character	RRX.LOCATION	RRX-4
2	Decimal	RRX.ROAD-WIDTH	RRX-5
1	Decimal	RRX.DIRECTION-1	RRX-6
3	Decimal	RRX.ANGLE-OR1	RRX-7
4	Decimal	RRX.SIGHT-DIST-LEFT1	RRX-8
4	Decimal	RRX.SIGHT-DIST-RIGHT1	RRX-8
3	Decimal	RRX.DISTANCE-1	RRX-9
2	Decimal	RRX.CURVATURE-DEG-1	RRX-1Ø
2	Decimal	RRX.CURVATURE-MIN-1	RRX-1Ø
2	Decimal	RRX.GRADE-1	RRX-11
1	Decimal	RRX.LOCAL-INTERFERENCE-1	RRX-12
1	Decimal	RRX.VERT-SIGHT-RESTR-1	RRX-13
1	Decimal	RRX.DIRECTION-2	RRX-6
3	Decimal	RRX.ANGLE-OR2	RRX-7
4	Decimal	RRX.SIGHT-DIST-LEFT2	RRX-8
4	Decimal	RRX.SIGHT-DIST-RIGHT2	RRX-8
3	Decimal	RRX.DISTANCE-2	RRX-9
2	Decimal	RRX.CURVATURE-DEG-2	RRX-1Ø
2	Decimal	RRX.CURVATURE-MIN-2	RRX-1Ø
2	Decimal	RRX.GRADE-2	RRX-11
1	Decimal	RRX.LOCAL-INTERFERENCE-2	RRX-12
1	Decimal	RRX.VERT-SIGHT-RESTR-2	RRX-13
1	Decimal	RRX.#-DAILY-PASSENGER	RRX-14
2	Decimal	RRX.DAILY-PASS-SPEED	RRX-15
2	Decimal	RRX.#-DAILY-FREIGHT	RRX-14
2	Decimal	RRX.DAILY-FREIGHT-SPEED	RRX-15
3	Decimal	RRX.#-DAILY-SWITCH	RRX-14
2	Decimal	RRX.DAILY-SWITCH-SPEED	RRX-15

Length	Format	Data Element Name	Described Under
1	Decimal	RRX.#-WEEKLY-PASSENGER	RRX-14
2	Decimal	RRX.WEEKLY-PASS-SPEED	RRX-15
2	Decimal	RRX.#-WEEKLY-FREIGHT	RRX-14
2	Decimal	RRX.WEEKLY-FREIGHT-SPEED	RRX-15
3	Decimal	RRX.#-WEEKLY-SWITCH	RRX-14
2	Decimal	RRX.WEEKLY-SWITCH-SPEED	RRX-15
1	Decimal	RRX.#-SEASONAL-PASSENGER	RRX-14
2	Decimal	RRX.SEASONAL-PASS-SPEED	RRX-15
2	Decimal	RRX.#-SEASONAL-FREIGHT	RRX-14
2	Decimal	RRX.SEASONAL-FREIGHT-SPE	RRX-15
3	Decimal	RRX.#-SEASONAL-SWITCH	RRX-14
2	Decimal	RRX.SEASONAL-SWITCH-SPEE	RRX-15
6	Date	RRX.INVENTORY-DATE	RRX-16
2	Decimal	RRX.INVENTORY-MONTH	RRX-16
2	Decimal	RRX.INVENTORY-DAY	RRX-16
2	Decimal	RRX.INVENTORY-YEAR	RRX-16
4	Character	RRX.OPERATING-RAILROAD	RRX-17
7	Character	RRX.CROSSING-ID	RRX-18
15	Character	RRX.BRANCH-LINE	RRX-19
(6,2)	Decimal	RRX.BRANCH-LINE-MILEPOST	RRX-2Ø
1	Decimal	RRX.#-MAIN-TRACKS	RRX-21
2	Decimal	RRX.#-OTHER-TRACKS	RRX-21
1	Decimal	RRX.OTHER-RR-USE	RRX-22
4	Character	RRX.OTHER-RR-NAME	RRX-23
1	Decimal	RRX.#-CROSSBUCKS	RRX-25
1	Decimal	RRX.#-STOP-SIGNS	RRX-25
1	Decimal	RRX.#-WIGWAGS	RRX-25
1	Decimal	RRX.#-FLASHING-LIGHTS	RRX-25
1	Decimal	RRX.#-GATES	RRX-25
1	Decimal	RRX.#-LANES	RRX-26
3	Decimal	RRX.#-NIGHTLY-TRAINS	RRX-27
4	Decimal	RRX.EST-ADT	RRX-28
6	Date	RRX.DATE-OF-UPDATE	RRX-29
2	Decimal	RRX.MONTH-OF-UPDATE	RRX-29
2	Decimal	RRX.DAY-OF-UPDATE	RRX-29
2	Decimal	RRX.YEAR-OF-UPDATE	RRX-29

RRX-1 - Key - The format of the key field is:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L,M)
2-6	5	Route number
7-15	9	Milepoint in format 'nnn+n.nnn'

The data element names used with the key field are:

Data Element Name	Columns	Length
RRX.KEY	1-15	15
RRX.ROUTE-SYS-&-NUMBER	1-6	6
RRX.ROUTE-SYSTEM	1	1
RRX.ROUTE-NUMBER	2-6	5

Data Element Name	Columns	Length
RRX.MILEPOINT	7-15	9
RRX.REFERENCE-POST	7-9	3
RRX.DISTANCE	1Ø-15	6

RRX-2 - Date of Field Survey - The data element names are:

Data Element Name	Example
RRX.SURVEY-DATE	RRX.SURVEY-DATE *GE* 1/1/76
RRX.SURVEY-MONTH	RRX.SURVEY-MONTH *EQ* 6
RRX.SURVEY-DAY	RRX.SURVEY-DAY *GE* 15
RRX.SURVEY-YEAR	RRX.SURVEY-YEAR *GE* 76

RRX-3 - Urban/Rural Code - Use RRX.URBAN-RURAL *EQ* 'U' for urban crossings and RRX.URBAN-RURAL *EQ* 'R' for rural crossings.

RRX-4 - Verbal Description of Location - RRX.LOCATION can be used to reference the 25-character description field.

RRX-5 - Road Width - RRX.ROAD-WIDTH references the roadlog width at the crossing, and ranges from Ø to 99 feet. Example: RRX.ROAD-WIDTH *GE* 2Ø.

RRX-6 - Direction - RRX.DIRECTION-1 for road approach 1 and RRX.DIRECTION-2 for road approach 2. Codes are:

1	North	5	South
2	Northwest	6	Southeast
3	West	7	East
4	Southwest	8	Northeast

Example: RRX.DIRECTION-1 *EQ* 3.

RRX-7 - Angle at O - Use RRX.ANGLE-OR1 for road approach 1 and RRX.ANGLE-OR2 for road approach 2. The field contains a 3-digit angle in degrees. Example: RRX.ANGLE-OR1 *GE* 9Ø.

RRX-8 - Sight Distance - Use RRX.SIGHT-DISTANCE-LEFT1 and RRX.SIGHT-DISTANCE-RIGHT1 for road approach 1 and RRX.SIGHT-DISTANCE-LEFT2 and RRX.SIGHT-DISTANCE-RIGHT2 for road approach 2. Sight distances range from Ø to 9999 feet. Example: RRX.SIGHT-DISTANCE-LEFT1 *LE* 1ØØ.

RRX-9 - Distance OP - Use RRX.DISTANCE-1 for road approach 1 and RRX.DISTANCE-2 for road approach 2. Distances range from Ø to 999 feet. Example: RRX.DISTANCE-1 *LT* 1ØØ.

RRX-1Ø - Curvature - Use RRX.CURVATURE-DEG-1 and RRX.CURVATURE-MIN-1 for road approach 1 and RRX.CURVATURE-DEG-2 and RRX.CURVATURE-MIN-2 for road approach 2. Example: RRX.CURVATURE-DEG-1 *LT* 2Ø.

RRX-11 - Grade - Use RRX.GRADE-1 for road approach 1 and RRX.GRADE-2 for road approach 2. The field contains percentage of grade and is positive for increasing grade and negative for decreasing grade. Example: RRX.GRADE-1 *LT* -1.

RRX-12 - Local interference - Use RRX.LOCAL-INTERFERENCE-1 for road approach 1 and RRX.LOCAL-INTERFERENCE-2 for road approach 2. Codes are:

Ø	None	2	Intersecting street - no traffic controls
1	Busy driveway	3	Intersecting street - traffic controls

Example: RRX.LOCAL-INTERFERENCE-2 *EQ* 2.

RRX-13 - Vertical Sight Restrictions - Use RRX.VERT-SIGHT-RESTR-1 for road approach 1 and RRX.VERT-SIGHT-RESTR-2 for road approach 2. A value of zero indicates that it is possible to see a 4.5 foot vehicle on the other side of the crossing when 3ØØ feet back from the crossing. A value of one indicates that this is not possible. Example: RRX.VERT-SIGHT-RESTR *EQ* 1.

RRX-14 - Number of Trains - Use the following data element names:

Data Element Name	Digits	Examples
RRX.#-DAILY-PASSENGER	1	
RRX.#-WEEKLY-PASSENGER	1	RRX.#-WEEKLY-PASSENGER *GE* 3
RRX.#-SEASONAL-PASSENGER	1	
RRX.#-DAILY-FREIGHT	2	
RRX.#-WEEKLY-FREIGHT	2	RRX.#-WEEKLY-FREIGHT *GE* 8
RRX.#-SEASONAL-FREIGHT	2	
RRX.#-DAILY-SWITCH	3	
RRX.#-WEEKLY-SWITCH	3	RRX.#-WEEKLY-SWITCH *LE* 15
RRX.#-SEASONAL-SWITCH	3	

RRX-15 - Train Speeds - Use the following data element names:

Data Element Name	
RRX.DAILY-PASS-SPEED	
RRX.WEEKLY-PASS-SPEED	
RRX.SEASONAL-PASS-SPEED	Example: RRX.DAILY-PASS-SPEED *GE* 45.
RRX.DAILY-FREIGHT-SPEED	
RRX.WEEKLY-FREIGHT-SPEED	
RRX.SEASONAL-FREIGHT-SPE	
RRX.DAILY-SWITCH-SPEED	
RRX.WEEKLY-SWITCH-SPEED	
RRX.SEASONAL-SWITCH-SPEE	

RRX-16 - Date of DOT-AAR Crossing Inventory - Use the following data element names:

Data Element Name	Example
RRX.INVENTORY-DATE	RRX.INVENTORY-DATE *GE* 1/1/76
RRX.INVENTORY-DAY	
RRX.INVENTORY-MONTH	
RRX.INVENTORY-YEAR	RRX.INVENTORY-YEAR *GE* 76

RRX-17 - Operating Railroad - The data element name is RRX.OPERATING-RAILROAD. The name is stored using the FRA code, and is 4 characters in length.

RRX-18 - Crossing ID - The data element name is RRX.CROSSING-ID. Crossing ID numbers are 7 characters in length.

RRX-19 - Branch or Line Name - The data element name is RRX.BRANCH-LINE. The branch/line names are 15 characters in length.

RRX-2Ø - Branch or Line Milepost - The data element name is RRX.BRANCH-LINE-MILEPOST. The field is stored in nnnn.nn format. Example: RRX.BRANCH-LINE-MILEPOST *GE* 455.3.

RRX-21 - Number of Tracks - The data element names are RRX.#-MAIN-TRACKS (ranges from Ø to 9) and RRX.#-OTHER-TRACKS (ranges from Ø to 99). Example: RRX.#-OTHER-TRACKS *GE* 5.

RRX-22 - Type of Use - The data element name is RRX.OTHER-RR-USE. The codes are:

Ø Only operating RR uses tracks
1 Another RR operates separate tracks
2 Another RR operates shared tracks

Example: RRX.OTHER-RR-USE *EQ* 2.

RRX-23 - Other Railroad Name - The data element name is RRX.OTHER-RR-NAME. The field is like RRX-17 above.

RRX-25 - Signalization - Each of these fields contains a value ranging from Ø to 9:

Data Element Name	
RRX.#-CROSSBUCKS	
RRX.#-STOP-SIGNS	
RRX.#-WIGWAGS	Example: RRX.#-GATES *GE* 4.
RRX.#-FLASHING-LIGHTS	
RRX.#-GATES	

RRX-26 - Number of Traffic Lanes - RRX.#-LANES contains the number of highway lanes (Ø through 9) at the crossing. Example: RRX.#-LANES *GE* 4.

RRX-27 - Nightly Train Traffic - RRX.#-NIGHTLY-TRAINS contains the average number of nightly trains (Ø through 999). Example: RRX.#-NIGHTLY-TRAINS *GE* 4Ø.

RRX-28 - Estimated ADT - RRX.EST-ADT contains an estimated ADT at those crossings for which the traffic file does not contain traffic counts. The ADT can range from Ø to 9999. Example: RRX.EST-ADT *GT* 5ØØ.

RRX-29 - Date of Update - The data element names are RRX.DATE-OF-UPDATE, RRX.DAY-OF-UPDATE, RRX.MONTH-OF-UPDATE, and RRX.YEAR-OF-UPDATE. Examples: RRX.DATE-OF-UPDATE *GE* 1/1/76, RRX.YEAR-OF-UPDATE *GE* 76.

Roadlog File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
2	Decimal	RLG.AREA-NAME	RLG-29
(3,1)	Decimal	RLG.BASE-THICKNESS	RLG-2Ø
3/18	Dec/Char	RLG.CITY	RLG-9
4	Character	RLG.COIN-DESCRIPTION	RLG-4
9	Character	RLG.COIN-MILEPOINT-1	RLG-4
9	Character	RLG.COIN-MILEPOINT-2	RLG-4
5	Character	RLG.COIN-ROUTE	RLG-4
6	Character	RLG.COIN-SYS-&-ROUTE	RLG-4
1	Character	RLG.SYSTEM	RLG-4
(5,3)	Decimal	RLG.CONSTRUCTED-LENGTH	RLG-3
2	Decimal	RLG.CONSTRUCTION-DIV	RLG-27
1	Decimal	RLG.CONTROL-OF-ACCESS	RLG-21
2/15	Dec/Char	RLG.COUNTY	RLG-11
35	Character	RLG.DESCRIPTION	RLG-4
6	Character	RLG.DISTANCE	RLG-1
1	Character	RLG.DIVIDED-CODE	RLG-6
6	Date	RLG.EFFECTIVE-DATE	RLG-25
2	Decimal	RLG.EFFECTIVE-DAY	RLG-25
2	Decimal	RLG.EFFECTIVE-MONTH	RLG-25
2	Decimal	RLG.EFFECTIVE-YEAR	RLG-25
2	Decimal	RLG.FINANCIAL-DISTRICT	RLG-12
2	Decimal	RLG.FOREST-HIGHWAY-NMBR	RLG-15
1	Decimal	RLG.FUNCTIONAL-CLASS	RLG-26
2	Decimal	RLG.JURISDICTION	RLG-16
15	Character	RLG.KEY	RLG-1
2	Decimal	RLG.LOCATION-CODE-1	RLG-17
2	Decimal	RLG.LOCATION-CODE-2	RLG-17
4	Decimal	RLG.MAINT-SECTION	RLG-24
2	Decimal	RLG.MAINTENANCE-DIV	RLG-18
2	Decimal	RLG.MAP-SHEET	RLG-3Ø
9	Character	RLG.MILEPOINT	RLG-1
1	Decimal	RLG.NUMBER-OF-LANES	RLG-7
1	Decimal	RLG.ONE-WAY-CODE	RLG-8
2	Decimal	RLG.PLANNING-DIVISION	RLG-28
1	Decimal	RLG.POPULATION	RLG-1Ø
2/4	Dec/Char	RLG.PROJECT-CLASS	RLG-5
11	Character	RLG.PROJECT-NUMBER	RLG-5
3	Character	RLG.RANGE	RLG-31
3	Character	RLG.REFERENCE-POST	RLG-1
2	Character	RLG.REMARK	RLG-2
2	Decimal	RLG.ROADWAY-WIDTH	RLG-19
(5,3)	Decimal	RLG.ROUTE-LENGTH	RLG-3
5	Character	RLG.ROUTE-NUMBER	RLG-1
6	Character	RLG.ROUTE-SYS-&-NUMBER	RLG-1
1	Character	RLG.ROUTE-SYSTEM	RLG-1

Length	Format	Data Element Name	Described Under
2	Character	RLG.SECTION	RLG-31
(5,3)	Decimal	RLG.SECTION-LENGTH	RLG-3
8	Character	RLG.SECTN-TOWNSHIP-RANGE	RLG-31
(2,1)	Decimal	RLG.SURFACE-THICKNESS	RLG-2Ø
4	Decimal	RLG.SURFACE-TYPE	RLG-22
1	Decimal	RLG.SURFACE-TYPE-CLASS	RLG-23
2	Decimal	RLG.SURFACE-WIDTH	RLG-19
3	Character	RLG.TOWNSHIP	RLG-31
(5,3)	Decimal	RLG.UNIMPROVED-LENGTH	RLG-3
6	Date	RLG.UPDATE-DATE	RLG-32
2	Decimal	RLG.UPDATE-DAY	RLG-32
2	Decimal	RLG.UPDATE-MONTH	RLG-32
2	Decimal	RLG.UPDATE-YEAR	RLG-32
(3,3)	Decimal	RLG.WYE-LENGTH	RLG-3
2	Decimal	RLG.YEAR-BUILT	RLG-13
2	Decimal	RLG.YEAR-IMPROVED	RLG-14

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
15	Character	RLG.KEY	RLG-1
6	Character	RLG.ROUTE-SYS-&-NUMBER	RLG-1
1	Character	RLG.ROUTE-SYSTEM	RLG-1
5	Character	RLG.ROUTE-NUMBER	RLG-1
9	Character	RLG.MILEPOINT	RLG-1
3	Character	RLG.REFERENCE-POST	RLG-1
6	Character	RLG.DISTANCE	RLG-1
2	Character	RLG.REMARK	RLG-2
(5,3)	Decimal	RLG.SECTION-LENGTH	RLG-3
(5,3)	Decimal	RLG.ROUTE-LENGTH	RLG-3
(5,3)	Decimal	RLG.CONSTRUCTED-LENGTH	RLG-3
(5,3)	Decimal	RLG.UNIMPROVED-LENGTH	RLG-3
(3,3)	Decimal	RLG.WYE-LENGTH	RLG-3
35	Character	RLG.DESCRIPTION	RLG-4
4	Character	RLG.COIN-DESCRIPTION	RLG-4
6	Character	RLG.COIN-SYS-&-ROUTE	RLG-4
1	Character	RLG.COIN-SYSTEM	RLG-4
5	Character	RLG.COIN-ROUTE	RLG-4
9	Character	RLG.COIN-MILEPOINT-1	RLG-4
9	Character	RLG.COIN-MILEPOINT-2	RLG-4
2/4	Dec/Char	RLG.PROJECT-CLASS	RLG-5
11	Character	RLG.PROJECT-NUMBER	RLG-5
1	Character	RLG.DIVIDED-CODE	RLG-6
1	Decimal	RLG.NUMBER-OF-LANES	RLG-7
1	Decimal	RLG.ONE-WAY-CODE	RLG-8
3/18	Dec/Char	RLG.CITY	RLG-9
1	Decimal	RLG.POPULATION	RLG-1Ø
2/15	Dec/Char	RLG.COUNTY	RLG-11
2	Decimal	RLG.FINANCIAL-DISTRICT	RLG-12

Length	Format	Data Element Name	Described Under
2	Decimal	RLG.YEAR-BUILT	RLG-13
2	Decimal	RLG.YEAR-IMPROVED	RLG-14
2	Decimal	RLG.FOREST-HIGHWAY-NMBR	RLG-15
2	Decimal	RLG.JURISDICTION	RLG-16
2	Decimal	RLG.LOCATION-CODE-1	RLG-17
2	Decimal	RLG.LOCATION-CODE-2	RLG-17
2	Decimal	RLG.MAINTENANCE-DIV	RLG-18
2	Decimal	RLG.SURFACE-WIDTH	RLG-19
2	Decimal	RLG.ROADWAY-WIDTH	RLG-19
(2,1)	Decimal	RLG.SURFACE-THICKNESS	RLG-2Ø
(3,1)	Decimal	RLG.BASE-THICKNESS	RLG-2Ø
1	Decimal	RLG.CONTROL-OF-ACCESS	RLG-21
4	Decimal	RLG.SURFACE-TYPE	RLG-22
1	Decimal	RLG.SURFACE-TYPE-CLASS	RLG-23
4	Decimal	RLG.MAINT-SECTION	RLG-24
6	Date	RLG.EFFECTIVE-DATE	RLG-25
2	Decimal	RLG.EFFECTIVE-DAY	RLG-25
2	Decimal	RLG.EFFECTIVE-MONTH	RLG-25
2	Decimal	RLG.EFFECTIVE-YEAR	RLG-25
1	Decimal	RLG.FUNCTIONAL-CLASS	RLG-26
2	Decimal	RLG.CONSTRUCTION-DIV	RLG-27
2	Decimal	RLG.PLANNING-DIVISION	RLG-28
2	Decimal	RLG.AREA-NAME	RLG-29
2	Decimal	RLG.MAP-SHEET	RLG-3Ø
8	Character	RLG.SECTN-TOWNSHIP-RANGE	RLG-31
2	Character	RLG.SECTION	RLG-31
3	Character	RLG.TOWNSHIP	RLG-31
3	Character	RLG.RANGE	RLG-31
6	Date	RLG.UPDATE-DATE	RLG-32
2	Decimal	RLG.UPDATE-MONTH	RLG-32
2	Decimal	RLG.UPDATE-DAY	RLG-32
2	Decimal	RLG.UPDATE-YEAR	RLG-32

RLG-1 - Key - The key has the following format:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L)
2-6	5	Route number
7-15	9	Milepoint in format 'nnn+n.nnn'

Use the following data element names:

Data Element Name	Columns	Length
RLG.KEY	1-15	15
RLG.ROUTE-SYS-&-NUMBER	1-6	6
RLG.ROUTE-SYSTEM	1	1
RLG.ROUTE-NUMBER	2-6	5
RLG.MILEPOINT	7-15	9
RLG.REFERENCE-POST	7-9	3
RLG.DISTANCE	1Ø-15	6

RLG-2 - Remark - The data element name is RLG.REMARK. The remark codes are:

' '	"Normal"	'DS'	Description
'SP'	Spur	'ER'	Description
'LP'	Loop	'EN'	End of route
'NE'	Non-existent	'CO'	Coincident
'OS'	Out of state	'IL'	Interstate loop

Example: RLG.REMARK *EQ* 'EN'.

RLG-3 - Lengths - All lengths except wye length range from .ØØØ to 99.999 miles. Wye length ranges from .ØØØ to .999 miles. Data element names are:

RLG.SECTION-LENGTH
RLG.ROUTE-LENGTH
RLG.CONSTRUCTED-LENGTH
RLG.UNIMPROVED-LENGTH
RLG.WYE-LENGTH

Examples:

RLG.SECTION-LENGTH *GE* 1.

RLG.ROUTE-LENGTH *LE* 1.53.

RLG-4 - Description - Most records contain a 35-character free-format description. CO and IL records contain a description in the following format:

Columns	Length	Contents
1-4	4	'COIN' (CO) or 'LOOP' (IL)
5	1	' '
6	1	Route system (I,P,S,U,L)
7-11	5	Route number
12	1	' '
13-21	9	Starting milepoint
22	1	'-'
23-31	9	Ending milepoint
32-35	4	' '

Use the following data element names:

Data Element Name	Columns	Length
RLG.DESCRIPTION	1-35	35
RLG.COIN-DESCRIPTION	1-4	4
RLG.COIN-SYS-&-ROUTE	6-11	6
RLG.COIN-SYSTEM	6	1
RLG.COIN-ROUTE	7-11	5
RLG.COIN-MILEPOINT-1	13-21	9
RLG.COIN-MILEPOINT-2	23-31	9

Example: RLG.COIN-DESCRIPTION *EQ* 'COIN' *AND* RLG.COIN-SYSTEM *EQ* 'I'.

RLG-5 - Project Number - RLG.PROJECT-NUMBER refers to the entire 11-character project number. RLG.PROJECT-CLASS refers only to the project classification portion of the number. Project classifications can be specified as the alphabetic project class (eg., RLG.PROJECT-CLASS *EQ* 'FAGS') or as the numeric administration code (eg., RLG.PROJECT-CLASS *EQ* 4Ø). Project class codes and administration codes can be found on page 5-13 of the data coding manual.

RLG-6 - Divided Code - Use RLG.DIVIDED-CODE *EQ* 'D' to locate divided sections and RLG.DIVIDED-CODE *NE* 'D' to locate undivided sections.

RLG-7 - Number of Lanes - The data element name is RLG.NUMBER-OF-LANES. The field contains a number ranging from Ø to 9. Example: RLG.NUMBER-OF-LANES *GE* 4.

RLG-8 - One-Way Code - RLG.ONE-WAY-CODE contains one of the following codes:

Ø Not stated
1 One-way traffic Example: RLG.ONE-WAY-CODE *EQ* 1.
2 Two-way traffic

RLG-9 - City Name or Number - RLG.CITY contains the 3-digit city number. Rural section contain zero in RLG.CITY. You may specify the city name (eg., RLG.CITY *EQ* 'GREAT-FALLS') or the city number (eg., RLG.CITY *EQ* 52). The city names and numbers are shown in table 2-1 of chapter 2.

RLG-1Ø - Population - RLG.POPULATION contains one of the following codes:

Ø	Rural	4	5,ØØØ - 9,999
1	Ø - 999	5	1Ø,ØØØ - 24,999
2	1,ØØØ - 2,499	6	25,ØØØ - 49,999
3	2,5ØØ - 4,999	7	5Ø,ØØØ and over

Example: RLG.POPULATION *GE* 4.

RLG-11 - County Name or Number - RLG.COUNTY contains the 2-digit county number. You may specify counties by name (eg., RLG.COUNTY *EQ* 'SWEET-GRASS') or by number (eg., RLG.COUNTY *EQ* 49). If you use county number, specify the alphabetical county number. The county names and numbers are shown in table 5-1 of chapter 5.

RLG-12 - Financial District - RLG.FINANCIAL-DISTRICT contains a value ranging from 1 to 12 in any record that has a non-zero county number and a zero in all other records. Example: RLG.FINANCIAL-DISTRICT *EQ* 5.

RLG-13 - Year Built - RLG.YEAR-BUILT contains the year built. Example: RLG.YEAR-BUILT *LE* 5Ø.

RLG-14 - Year Improved - RLG.YEAR-IMPROVED contains the year improved. Example: RLG.YEAR-IMPROVED *LE* 55.

RLG-15 - Forest Highway Number - RLG.FOREST-HIGHWAY-NMBR contains the forest highway number (zero is stored for non-forest highway sections). Example: RLG.FOREST-HIGHWAY-NMBR *NE* Ø.

RLG-16 - Jurisdiction - RLG.JURISDICTION contains a 2-digit jurisdiction code. The jurisdiction codes are:

Ø1	County	Ø6	National wildlife refuge
Ø2	National forest	Ø7	Bureau of Land Management
Ø3	Indian reservation	Ø8	State park
Ø4	National park	Ø9	State forest
Ø5	National monument	12	State game preserve

Example: RLG.JURISDICTION *EQ* 9.

RLG-17 - Location Codes - Two fields, RLG.LOCATION-CODE-1 and RLG.LOCATION-CODE-2, are used for location codes. When both are coded, the first field contains the smaller value. Otherwise, the second code contains zero. The location codes are:

Ø1	City	Ø8	National wildlife refuge
Ø2	Urban (outside city)	Ø9	Military reservation
Ø3	County	1Ø	State forest
Ø4	Indian reservation	11	State park
Ø5	National forest	12	State game preserve
Ø6	National monument	13	Bureau of Land Management
Ø7	National park		

Example: RLG.LOCATION-CODE-1 *EQ* 8 *OR* RLG.LOCATION-CODE-2 *EQ* 8.

RLG-18 - Maintenance Division - RLG.MAINTENANCE-DIV contains one of the following maintenance division numbers:

11	21	31	41	51
12	22	32	42	52
				53

Example: RLG.MAINTENANCE-DIV *EQ* 42.

RLG-19 - Width - RLG.SURFACE-WIDTH contains the width excluding shoulders and RLG.ROADWAY-WIDTH contains the width including shoulders. Both range from Ø to 99 feet. Example: RLG.SURFACE-WIDTH *EQ* 4Ø.

RLG-2Ø - Thickness - RLG.SURFACE-THICKNESS (Ø to 9.9 inches) and RLG.BASE-THICKNESS (Ø to 99.9 inches) contains the surface and base thicknesses. Example: RLG.BASE-THICKNESS *LE* 5.8.

RLG-21 - Control of Access - RLG.CONTROL-OF-ACCESS contains one of the following codes:

Ø Not stated or not applicable
1 No control
2 Partial control
3 Full control

Example: RLG.CONTROL-OF-ACCESS *EQ* 3.

RLG-22 - Surface Type - RLG.SURFACE-TYPE contains the 4-digit surface type classification. The surface type codes can be found on page 5-2Ø of the data coding manual. Example: RLG.SURFACE-TYPE *EQ* 321Ø.

RLG-23 - Surface Type Classification - RLG.SURFACE-TYPE-CLASS contains one of the following codes:

1	Primitive	5	Bituminous surface treatment
2	Unimproved	6	Road mix
3	Graded and drained	7	Plant mix
4	Gravel	8	Portland cement concrete

Example: RLG.SURFACE-TYPE-CLASS *EQ* 4.

RLG-24 - Maintenance Section - RLG.MAINT-SECTION contains the 4-digit maintenance section number. Example: RLG.MAINT-SECTION *EQ* 11Ø1.

RLG-25 - Effective Date - The data element names are RLG.EFFECTIVE-DATE, RLG.EFFECTIVE-MONTH, RLG.EFFECTIVE-DAY, and RLG.EFFECTIVE-YEAR. Examples: RLG.EFFECTIVE-DATE *GE* 4/7/76, RLG.EFFECTIVE-YEAR *GE* 76.

RLG-26 - Functional Classification - RLG.FUNCTIONAL-CLASS contains one of the following codes:

1	Interstate	4	Major collector
2	Principal arterial	5	Minor collector
3	Minor arterial	6	Local road

Example: RLG.FUNCTIONAL-CLASS *GE* 4.

RLG-27 - Construction Division - RLG.CONSTRUCTION-DIV contains the number of a construction division (1 to 12).

RLG-28 - Planning Division - RLG.PLANNING-DIVISION contains the number of a planning division (1 to 12).

RLG-29 - Area Name - RLG.AREA-NAME contains a 2-digit area name code. The area names can be found on page 5-26 of the data coding manual. Example: RLG.AREA-NAME *EQ* 84.

RLG-3Ø - Map Sheet - RLG.MAP-SHEET contains 1 2-digit map sheet number.

RLG-31 - Range, Township, and Section - Use RLG.SECTN-TOWNSHIP-RANGE to refer to the entire field. Use RLG.SECTION, RLG.RANGE, or RLG.TOWNSHIP to refer to subfields.

RLG-32 - Date of Update - The data element names are RLG.UPDATE-DATE, RLG.UPDATE-DAY, RLG.UPDATE-MONTH, and RLG.UPDATE-YEAR. Examples: RLG.UPDATE-DATE *GE* 4/9/77, RLG.UPDATE-YEAR *GE* 77.

Skid File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
3	Character	SKD.CITY	SKD-2
38	Character	SKD.COMMENTS	SKD-8
6	Date	SKD.DATE-OF-UPDATE	SKD-14
2	Character	SKD.DAY-OF-TEST	SKD-2
2	Decimal	SKD.DAY-OF-UPDATE	SKD-14
3	Character	SKD.DIST-FROM-INTSECTN	SKD-2
6	Character	SKD.DISTANCE	SKD-2
1	Decimal	SKD.FOREIGN-MATTER	SKD-1Ø
1	Character	SKD.GRADE	SKD-5
1	Character	SKD.HORIZ-CURVE	SKD-4
28	Character	SKD.KEY	SKD-2
1	Character	SKD.LANE	SKD-2
1	Character	SKD.LOCATION	SKD-6
9	Character	SKD.MILEPOINT	SKD-2
2	Character	SKD.MONTH-OF-TEST	SKD-2
2	Decimal	SKD.MONTH-OF-UPDATE	SKD-14
32	Character	SKD.MUNIC-COMMENTS	SKD-8
1	Character	SKD.MUNIC-DIRECTION	SKD-2
5	Character	SKD.MUNIC-ROUTE-NUMBER	SKD-8
1	Character	SKD.MUNIC-ROUTE-SYSTEM	SKD-8
6	Character	SKD.MUNIC-SYS-&-ROUTE	SKD-8
3	Decimal	SKD.PAVEMENT-TEMP	SKD-11
1	Character	SKD.RECORD-TYPE	SKD-1
3	Character	SKD.REFERENCE-POST	SKD-2
5	Character	SKD.ROUTE-NUMBER	SKD-2
6	Character	SKD.ROUTE-SYS-&-NUMBER	SKD-2
1	Character	SKD.ROUTE-SYSTEM	SKD-2
1	Character	SKD.RURAL-DIRECTION	SKD-2
4	Character	SKD.SEQUENCE-NUMBER	SKD-2
2	Decimal	SKD.SKID-NUMBER	SKD-13
2	Decimal	SKD.SPEED	SKD-12
1	Character	SKD.SURFACE-REPAIR	SKD-7
1	Decimal	SKD.SURFACE-TEXTURE	SKD-9
1	Character	SKD.SURFACE-TYPE	SKD-3
1	Character	SKD.WHEEL	SKD-2
4	Character	SKD.X-COORDINATE	SKD-2
4	Character	SKD.Y-COORDINATE	SKD-2
2	Character	SKD.YEAR-OF-TEST	SKD-2
2	Decimal	SKD.YEAR-OF-UPDATE	SKD-14

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
1	Character	SKD.RECORD-TYPE	SKD-1
28	Character	SKD.KEY	SKD-2
1	Character	SKD.ROUTE-SYSTEM	SKD-2
5	Character	SKD.ROUTE-NUMBER	SKD-2
6	Character	SKD.ROUTE-SYS-&-NUMBER	SKD-2
9	Character	SKD.MILEPOINT	SKD-2
3	Character	SKD.REFERENCE-POST	SKD-2
6	Character	SKD.DISTANCE	SKD-2
1	Character	SKD.RURAL-DIRECTION	SKD-2
3	Character	SKD.CITY	SKD-2
4	Character	SKD.X-COORDINATE	SKD-2
4	Character	SKD.Y-COORDINATE	SKD-2
1	Character	SKD.MUNIC-DIRECTION	SKD-2
3	Character	SKD.DIST-FROM-INTSECTN	SKD-2
1	Character	SKD.LANE	SKD-2
1	Character	SKD.WHEEL	SKD-2
2	Character	SKD.MONTH-OF-TEST	SKD-2
2	Character	SKD.DAY-OF-TEST	SKD-2
2	Character	SKD.YEAR-OF-TEST	SKD-2
4	Character	SKD.SEQUENCE-NUMBER	SKD-2
1	Character	SKD.SURFACE-TYPE	SKD-3
1	Character	SKD.HORIZ-CURVE	SKD-4
1	Character	SKD.GRADE	SKD-5
1	Character	SKD.LOCATION	SKD-6
1	Character	SKD.SURFACE-REPAIR	SKD-7
38	Character	SKD.COMMENTS	SKD-8
6	Character	SKD.MUNIC-SYS-&-ROUTE	SKD-8
1	Character	SKD.MUNIC-ROUTE-SYSTEM	SKD-8
5	Character	SKD.MUNIC-ROUTE-NUMBER	SKD-8
32	Character	SKD.MUNIC-COMMENTS	SKD-8
1	Decimal	SKD.SURFACE-TEXTURE	SKD-9
1	Decimal	SKD.FOREIGN-MATTER	SKD-1Ø
3	Decimal	SKD.PAVEMENT-TAMP	SKD-11
2	Decimal	SKD.SPEED	SKD-12
2	Decimal	SKD.SKID-NUMBER	SKD-13
6	Date	SKD.DATE-OF-UPDATE	SKD-14
2	Decimal	SKD.MONTH-OF-UPDATE	SKD-14
2	Decimal	SKD.DAY-OF-UPDATE	SKD-14
2	Decimal	SKD.YEAR-OF-UPDATE	SKD-14

SKD-1 - Record Type - SKD.RECORD-TYPE contains an 'A' in those records converted from the earlier file.

SKD-2 - Key - The format of the key is different for rural and municipal tests. The rural format is:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L)
2-6	5	Route number
7-15	9	Milepoint in format 'nnn+n.nnn'
16	1	Direction (I,D,R)
17	1	Lane (I,M,O,L,R,A)
18	1	Wheel (L,R)
19-24	6	Date (month,day,year)
25-28	4	Sequence number

The municipal format is:

Columns	Length	Contents
1	1	'M'
2-4	3	City number
5-8	4	X-coordinate
9-12	4	Y-coordinate
13	1	Direction (N,S,E,W,Ø-9,R)
14-16	3	Distance from intersection ('ØØØ' - non-intersection)
17	1	Lane (I,M,O,L,R,A)
18	1	Wheel (L,R)
19-24	6	Date (month,day,year)
25-28	4	Sequence number

The data element names used with the key are:

Data Element Name	Type	Columns	Length
SKD.KEY	Both	1-28	28
SKD.ROUTE-SYS-&-NUMBER	Rural	1-6	6
SKD.ROUTE-NUMBER	Rural	2-6	5
SKD.ROUTE-SYSTEM	Both	1	1
SKD.REFERENCE-POST	Rural	7-9	3
SKD.DISTANCE	Rural	1Ø-15	6
SKD.MILEPOINT	Rural	7-15	9
SKD.RURAL-DIRECTION	Rural	16	1
SKD.CITY	Municipal	2-4	3
SKD.X-COORDINATE	Municipal	5-8	4
SKD.Y-COORDINATE	Municipal	9-12	4
SKD.MUNIC-DIRECTION	Municipal	13	1
SKD.DIST-FROM-INTSECTN	Municipal	14-16	3
SKD.LANE	Both	17	1
SKD.WHEEL	Both	18	1
SKD.MONTH-OF-TEST	Both	19-2Ø	2
SKD.DAY-OF-TEST	Both	21-22	2
SKD.YEAR-OF-TEST	Both	23-24	2
SKD.SEQUENCE-NUMBER	Both	25-28	4

Example: SKD.ROUTE-SYSTEM *EQ* 'M' *AND* SKD.CITY *EQ* 'Ø52'

Note: See table 2-1 in chapter 2 for list of city numbers.

SKD-3 - Surface Type - SKD.SURFACE-TYPE contains one of these codes:

'A' Asphalt
'C' Concrete
'O' Other
' ' Not stated

Example: SKD.SURFACE-TYPE *EQ* 'O'

SKD-4 - Horizontal Curvature - SKD.HORIZ-CURVE contains one of these codes:

'T' Tangent
'C' Curve
' ' Not stated

Example: SKD.HORIZ-CURVE *EQ* 'C'

SKD-5 - Grade - SKD.GRADE contains one of these codes:

'N' None or negligible
'U' Upgrade
'D' Downgrade
' ' Not stated

Example: SKD.GRADE *EQ* 'U'

SKD-6 - Location - SKD.LOCATION contains one of these codes:

'N' Neither at intersection approach nor on a bridge
'B' Bridge
'I' Intersection approach
'C' Combination (both)
' ' Not stated

Example: SKD.LOCATION *EQ* 'C'

SKD-7 - Surface Repair - SKD.SURFACE-REPAIR contains one of these codes:

'N' Neither short overlay nor spot patches
'O' Short overlay section
'P' Spot patches
' ' Not stated

Example: SKD.SURFACE-REPAIR *EQ* 'P'

SKD-8 - Comments - For rural records, the comments field is a 38-character verbal description. For municipal records, the comments field consists of a 1-character route system, a 5-character route number, and a 32-character verbal description. The data element names are:

Data Element Name	Columns	Length
SKD.COMMENTS	1-38	38
SKD.MUNIC-SYS-&-ROUTE	1-6	6
SKD.MUNIC-ROUTE-SYSTEM	1	1
SKD.MUNIC-ROUTE-NUMBER	2-6	5
SKD.MUNIC-COMMENTS	7-38	32

Example: SKD.ROUTE-SYSTEM *EQ* 'M' *AND* SKD.MUNIC-ROUTE-SYSTEM *EQ* 'P'.

SKD-9 - Surface Texture - SKD.SURFACE-TEXTURE contains one of these codes:

Code	Concrete	Asphalt
Ø	Not stated	Not stated
1	Broomed or rough	Coarse textured knobby or angular
2	Normal concrete	Normal fine textured
3	Polished	Polished aggregate
4	Stud-worn	Flushed, bleeding, or overly rich
5	Other	Other

Example: SKD.SURFACE-TYPE *EQ* 'C' *AND* SKD.SURFACE-TEXTURE *EQ* 4.

SKD-1Ø - Foreign Matter - SKD.FOREIGN-MATTER contains one of these codes:

1 None
2 Loose aggregate/sand
3 Dirt or mud
4 Leaves
5 Garbage/litter
6 Manure
7 Farm products (eg., spilled grain)
8 Other
Ø Not stated

Example: SKD.FOREIGN-MATTER *GE* 2.

SKD-11 - Pavement Temperature - SKD.PAVEMENT-TEMP contains the pavement temperature in degrees farenheit and can range from Ø to 999. Example: SKD.PAVEMENT-TEMP *GT* 8Ø.

SKD-12 - Speed - SKD.SPEED contains the speed (in mph) the test vehicle was traveling, and can range from Ø to 99. Example: SKD.SPEED *NE* 4Ø.

SKD-13 - Skid Number - SKD.SKID-NUMBER contains the skid number (Ø to 99) of the test. Example: SKD.SKID-NUMBER *LT* 4Ø.

SKD-14 - Date of Update - The data element names are SKD.DATE-OF-UPDATE, SKD.MONTH-OF-UPDATE, SKD.DAY-OF-UPDATE, and SKD.YEAR-OF-UPDATE. Examples: SKD.DATE-OF-UPDATE *GE* 1/1/76, SKD.YEAR-OF-UPDATE *GE* 76.

Sufficiency File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
2	Decimal	SUF.#-CURVES	SUF-8
1	Decimal	SUF.#-NARROW-BRIDGES	SUF-9
2	Decimal	SUF.AVERAGE-SPEED	SUF-5
6	Date	SUF.DATE-OF-UPDATE	SUF-15
2	Decimal	SUF.DAY-OF-UPDATE	SUF-15
18	Character	SUF.DESCRIPTION	SUF-2
2	Decimal	SUF.DESIGN-SPEED	SUF-3
6	Character	SUF.DISTANCE	SUF-1
2	Decimal	SUF.DRAINAGE-RATING	SUF-12
6	Date	SUF.EFFECTIVE-DATE	SUF-14
2	Decimal	SUF.EFFECTIVE-DAY	SUF-14
2	Decimal	SUF.EFFECTIVE-MONTH	SUF-14
2	Decimal	SUF.EFFECTIVE-YEAR	SUF-14
2	Decimal	SUF.FOUNDATION-RATING	SUF-1Ø
15	Character	SUF.KEY	SUF-1
9	Character	SUF.MILEPOINT	SUF-1
2	Decimal	SUF.MONTH-OF-UPDATE	SUF-15
3	Character	SUF.REFERENCE-POST	SUF-1
5	Character	SUF.ROUTE-NUMBER	SUF-1
6	Character	SUF.ROUTE-SYS-&-NUMBER	SUF-1
1	Character	SUF.ROUTE-SYSTEM	SUF-1
(7,3)	Decimal	SUF.SECTION-LENGTH	SUF-13
2	Decimal	SUF.SIGHT-DISTANCE	SUF-6
2	Decimal	SUF.STOPPING-DISTANCE	SUF-7
2	Decimal	SUF.SURFACE-RATING	SUF-11
1	Decimal	SUF.TERRAIN	SUF-4
2	Decimal	SUF.YEAR-OF-UPDATE	SUF-15

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
15	Character	SUF.KEY	SUF-1
6	Character	SUF.ROUTE-SYS-&-NUMBER	SUF-1
1	Character	SUF.ROUTE-SYSTEM	SUF-1
5	Character	SUF.ROUTE-NUMBER	SUF-1
9	Character	SUF.MILEPOINT	SUF-1
3	Character	SUF.REFERENCE-POST	SUF-1
6	Character	SUF.DISTANCE	SUF-1
18	Character	SUF.DESCRIPTION	SUF-2
2	Decimal	SUF.DESIGN-SPEED	SUF-3
1	Decimal	SUF.TERRAIN	SUF-4
2	Decimal	SUF.AVERAGE-SPEED	SUF-5
2	Decimal	SUF.SIGHT-DISTANCE	SUF-6
2	Decimal	SUF.STOPPING-DISTANCE	SUF-7
2	Decimal	SUF.#-CURVES	SUF-8
1	Decimal	SUF.#-NARROW-BRIDGES	SUF-9

Length	Format	Data Element Name	Described Under
2	Decimal	SUF.FOUNDATION-RATING	SUF-1∅
2	Decimal	SUF.SURFACE-RATING	SUF-11
2	Decimal	SUF.DRAINAGE-RATING	SUF-12
(7,3)	Decimal	SUF.SECTION-LENGTH	SUF-13
6	Date	SUF.EFFECTIVE-DATE	SUF-14
2	Decimal	SUF.EFFECTIVE-MONTH	SUF-14
2	Decimal	SUF.EFFECTIVE-DAY	SUF-14
2	Decimal	SUF.EFFECTIVE-YEAR	SUF-14
6	Date	SUF.DATE-OF-UPDATE	SUF-15
2	Decimal	SUF.MONTH-OF-UPDATE	SUF-15
2	Decimal	SUF.DAY-OF-UPDATE	SUF-15
2	Decimal	SUF.YEAR-OF-UPDATE	SUF-15

SUF-1 - Key - The key is stored in the following format:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L)
2-6	5	Route number
7-15	9	Milepoint in format nnn+n.nnn

The following data element names can be used to reference the key:

Data Element Name	Columns	Length
SUF.KEY	1-15	15
SUF.ROUTE-SYS-&-NUMBER	1-6	6
SUF.ROUTE-SYSTEM	1	1
SUF.ROUTE-NUMBER	2-6	5
SUF.MILEPOINT	7-15	9
SUF.REFERENCE-POST	7-9	3
SUF.DISTANCE	1∅-15	6

SUF-2 - Description - SUF.DESCRIPTION contains an 18-character verbal description. The following special names indicate descriptor records:

'CITY'
'COINCIDENT'
'SPUR'
'LOOP'
'OUT-OF-STATE'
'END OF ROUTE'

SUF-3 - Design Speed - SUF.DESIGN-SPEED contains the 2-digit design speed. Two codes have special meanings:

∅∅ Non-existent section
∅1 Section under construction

Example: SUF.DESIGN-SPEED *GT* 1 *AND* SUF.DESIGN-SPEED *LT* 5∅.

SUF-4 - Terrain - SUF.TERRAIN contains one of the following codes:

1 Plains
2 Rolling
3 Mountainous

Example: SUF.TERRAIN *EQ* 3.

Note: SUF.TERRAIN contains zero in descriptor records and in records in which the design speed is ØØ or Ø1.

SUF-5 - Average Speed - SUF.AVERAGE-SPEED contains the 2-digit average speed (weighted average of design speeds). Example: SUF.AVERAGE-SPEED *GE* 5Ø.

SUF-6 - Percent of Sight Distance Less Than Design - SUF.SIGHT-DISTANCE contains the percentage of the section's length in which the sight distance is less than 15ØØ feet. Example: SUF.SIGHT-DISTANCE *GT* 5Ø.

SUF-7 - Number of Stopping Distances Less Than Design - SUF.STOPPING-DISTANCE contains the number of locations at which the stopping sight distance is inadequate. Example: SUF.STOPPING-DISTANCE *GE* 1Ø.

SUF-8 - Number of Sharp Curves - SUF.#-CURVES contains the number of curves in the section that are sharper than the design degree of curvature, and can range from Ø to 99. Example: SUF.#-CURVES *GE* 2Ø.

SUF-9 - Number of Narrow Bridges - SUF.#-NARROW-BRIDGES contains a count of the number of bridges in the section that are narrower than the traveled way. It can range from Ø to 9. Example: SUF.#-NARROW-BRIDGES *GE* 3.

SUF-1Ø - Foundation Rating - SUF.FOUNDATION-RATING contains either a Ø (inadequate) or a 1Ø (adequate). Example: SUF.FOUNDATION-RATING *EQ* 1Ø.

SUF-11 - Surface Rating - SUF.SURFACE-RATING contains a surface rating that ranges from Ø to 3Ø. Example: SUF.SURFACE-RATING *GE* 15.

SUF-12 - Drainage Rating - SUF.DRAINAGE-RATING contains a drainage rating that ranges from Ø to 1Ø. Example: SUF.DRAINAGE-RATING *GE* 8.

SUF-13 - Section Length - SUF.SECTION-LENGTH contains the section length for non-existent and under construction sections (see design speed above in SUF-3). A value of zero is stored for all other sections. Example: SUF.SECTION-LENGTH *NE* Ø. Note: the section length ranges from Ø to 9999.999.

SUF-14 - Effective Date - The effective date is referenced by the data elements SUF.EFFECTIVE-DATE, SUF.EFFECTIVE-MONTH, SUF.EFFECTIVE-DAY, and SUF.EFFECTIVE-YEAR. Examples: SUF.EFFECTIVE-DATE *GE* 4/1/76, SUF.EFFECTIVE-YEAR *GE* 76.

SUF-15 - Date of Update - The date of most recent update is referenced by the data elements SUF.DATE-OF-UPDATE, SUF.MONTH-OF-UPDATE, SUF.DAY-OF-UPDATE, and SUF.YEAR-OF-UPDATE. Examples: SUF.DATE-OF-UPDATE *GE* 4/1/76, SUF.YEAR-OF-UPDATE *GE* 76.

Traffic File

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
(3,3)	Decimal	TRF.%-COMMERCIAL-1 - 4	TRF-6
(3,3)	Decimal	TRF.%-OUT-OF-STATE-1 - 4	TRF-6
(3,3)	Decimal	TRF.%-PICKUPS-1 - 4	TRF-6
1	Character	TRF.ACTUAL-OR-ESTIMATED	TRF-2
5	Decimal	TRF.ADT-1 - 4	TRF-5
3	Decimal	TRF.DHV	TRF-8
6	Character	TRF.DISTANCE	TRF-1
6	Date	TRF.EFFECTIVE-DATE	TRF-9
2	Decimal	TRF.EFFECTIVE-DAY	TRF-9
2	Decimal	TRF.EFFECTIVE-MONTH	TRF-9
2	Decimal	TRF.EFFECTIVE-YEAR	TRF-9
3	Decimal	TRF.FUTURE-FACTOR	TRF-8
15	Character	TRF.KEY	TRF-1
9	Character	TRF.MILEPOINT	TRF-1
3	Character	TRF.REFERENCE-POST	TRF-1
1	Character	TRF.REMARK	TRF-3
5	Character	TRF.ROUTE-NUMBER	TRF-1
6	Character	TRF.ROUTE-SYS-&-NUMBER	TRF-1
1	Character	TRF.ROUTE-SYSTEM	TRF-1
6	Date	TRF.UPDATE-DATE	TRF-1Ø
2	Decimal	TRF.UPDATE-DAY	TRF-1Ø
2	Decimal	TRF.UPDATE-MONTH	TRF-1Ø
2	Decimal	TRF.UPDATE-YEAR	TRF-1Ø
2	Decimal	TRF.YEAR-1 - 4	TRF-5

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
15	Character	TRF.KEY	TRF-1
6	Character	TRF.ROUTE-SYS-&-NUMBER	TRF-1
5	Character	TRF.ROUTE-NUMBER	TRF-1
1	Character	TRF.ROUTE-SYSTEM	TRF-1
9	Character	TRF.MILEPOINT	TRF-1
3	Character	TRF.REFERENCE-POST	TRF-1
6	Character	TRF.DISTANCE	TRF-1
1	Character	TRF.ACTUAL-OR-ESTIMATED	TRF-2
1	Character	TRF.REMARK	TRF-3
2	Decimal	TRF.YEAR-1 - 4	TRF-4
5	Decimal	TRF.ADT-1 - 4	TRF-5
(3,3)	Decimal	TRF.%-OUT-OF-STATE-1 - 4	TRF-6
(3,3)	Decimal	TRF.%-PICKUPS-1 - 4	TRF-6
(3,3)	Decimal	TRF.%-COMMERCIAL-1 - 4	TRF-6
3	Decimal	TRF.FUTURE-FACTOR	TRF-7
3	Decimal	TRF.DHV	TRF-8

Length	Format	Data Element Name	Described Under
6	Date	TRF.EFFECTIVE-DATE	TRF-9
2	Decimal	TRF.EFFECTIVE-MONTH	TRF-9
2	Decimal	TRF.EFFECTIVE-DAY	TRF-9
2	Decimal	TRF.EFFECTIVE-YEAR	TRF-9
6	Date	TRF.UPDATE-DATE	TRF-1Ø
2	Decimal	TRF.UPDATE-MONTH	TRF-1Ø
2	Decimal	TRF.UPDATE-DAY	TRF-1Ø
2	Decimal	TRF.UPDATE-YEAR	TRF-1Ø

TRF-1 - Key - The key is stored in the following format:

Columns	Length	Contents
1	1	Route system (I,P,S,U,L)
2-6	5	Route number
7-15	9	Milepoint in format 'nnn+n.nnn'

The following data element names can be used to reference the key:

Data Element Name	Columns	Length
TRF.KEY	1-15	15
TRF.ROUTE-SYS-&-NUMBER	1-6	6
TRF.ROUTE-SYSTEM	1	1
TRF.ROUTE-NUMBER	2-6	5
TRF.MILEPOINT	7-15	9
TRF.REFERENCE-POST	7-9	3
TRF.DISTANCE	1Ø-15	6

TRF-2 - Actual or Estimated - TRF.ACTUAL-OR-ESTIMATED contains an 'A' for actual counts or an 'E' for estimated counts. Example: TRF.ACTUAL-OR-ESTIMATED *EQ* 'E'.

TRF-3 - Remark - TRF.REMARK contains one of the following codes:

'W' Rural major section break
'T' Municipal major section break
'N' Non-existent major section break
'O' Out-of-state major section break
'R' Rural minor section break
'M' Municipal minor section break
'C' Coincident break
'L' Loop break
'S' Spur break

Example: TRF.REMARK *EQ* 'R'.

TRF-4 - Year - TRF.YEAR-1 through TRF.YEAR-4 refer to the four year fields. TRF.YEAR-1 refers to the oldest year and TRF.YEAR-4 to the newest. Example: TRF.YEAR-1 *EQ* 74.

TRF-5 - ADT - TRF.ADT-1 through TRF.ADT-4 refer to the four ADT fields. TRF.ADT-1 refers to the oldest count and TRF.ADT-4 to the newest. Example: TRF.ADT-3 *GE* 1ØØØ. ADT's range from Ø to 99999.

TRF-6 - Percentages - TRF.%-OUT-OF-STATE-1 through TRF.%-OUT-OF-STATE-4 refer to the percentage of out of state vehicles. TRF.%-PICKUPS-1 through TRF.%-PICKUPS-4 refer to the percentage of pickups. TRF.%-COMMERCIAL-1 through TRF.%-COMMERCIAL-4 refer to the percentage of commercial vehicles. In each case, the first name refers to the oldest year and the fourth to the newest year. Values range from Ø to .999. Example: TRF.%-PICKUPS-3 *GE* .5.

TRF-7 - Future Factor - TRF.FUTURE-FACTOR contains a 3-digit future factor. Example: TRF.FUTURE-FACTOR *GE* 452.

TRF-8 - Design Hour Volume - TRF.DHV contains the 3-digit design hour volume. Example: TRF.DHV *GE* 4Ø2.

TRF-9 - Effective Date - The effective date is reference by the data elements TRF.EFFECTIVE-DATE, TRF.EFFECTIVE-MONTH, TRF.EFFECTIVE-DAY, and TRF.EFFECTIVE-YEAR. Examples: TRF.EFFECTIVE-DATE *GE* 4/1/76, TRF.EFFECTIVE-YEAR *GE* 76.

TRF-1Ø - Date of Update - The date of update is referenced by the data elements TRF.UPDATE-DATE, TRF.UPDATE-MONTH, TRF.UPDATE-DAY, and TRF.UPDATE-YEAR. Examples: TRF.UPDATE-DATE *GE* 4/1/76, TRF.UPDATE-YEAR *GE* 76.

Urban Sign Inventory Files

Listing of Names in Alphabetical Order

Length	Format	Data Element Name	Described Under
1	Decimal	USN.#-POSTS	USN-12
3	Decimal	USN.ASM:COST-1	USN-16
3	Decimal	USN.ASM:COST-2	USN-16
3	Decimal	USN.ASM:COST-3	USN-16
4	Decimal	USN.ASM:COST-4	USN-16
6	Date	USN.ASM:DATE	USN-3
2	Decimal	USN.ASM:DAY	USN-3
2	Decimal	USN.ASM:HOUR	USN-3
6	Date	USN.ASM:INS-DATE	USN-4
2	Decimal	USN.ASM:INS-DAY	USN-4
1	Character	USN.ASM:INS-INV-CODE	USN-4
2	Decimal	USN.ASM:INS-MONTH	USN-4
2	Decimal	USN.ASM:INS-YEAR	USN-4
2	Decimal	USN.ASM:MONTH	USN-3
3	Character	USN.ASM:SEQ-#	USN-1
6	Date	USN.ASM:UPDATE-DATE	USN-17
2	Decimal	USN.ASM:UPDATE-DAY	USN-17
2	Decimal	USN.ASM:UPDATE-MONTH	USN-17
2	Decimal	USN.ASM:UPDATE-YEAR	USN-17
4	Character	USN.ASM:UPDTE-CD	USN-2
2	Decimal	USN.ASM:YEAR	USN-3
5	Character	USN.ASMBLY-#	USN-1
14	Character	USN.ASMBLY-LOCATION	USN-7
1	Character	USN.ASMBLY-TYPE	USN-1Ø
2	Decimal	USN.CHNGE-LAT-CLR	USN-14
2	Decimal	USN.CHNGE-MTNG-HT	USN-14
2	Decimal	USN.COLOR	USN-28
4	Character	USN.CUR-SIGN-AGE	USN-35
1	Character	USN.DIR-TRAV	USN-7
1	Character	USN.DIRECTION-FACING	USN-21
5	Character	USN.DIST-FROM-INTSECT	USN-7
1	Decimal	USN.FACE	USN-31
1	Character	USN.FED-AID-SYS	USN-25
1	Decimal	USN.FUNCT-CLASS	USN-24
3	Decimal	USN.HORIZ-DIM	USN-33
2	Decimal	USN.LAT-CLR	USN-34
1	Decimal	USN.LETTER-TYPE	USN-29
1	Character	USN.MAINT-RESPN	USN-26
3	Character	USN.MAP-NUMBER	USN-9
1	Decimal	USN.MATERIAL	USN-3Ø
2	Decimal	USN.MTNG-HT	USN-34
1	Decimal	USN.POST-CONDITION	USN-5
1	Decimal	USN.POST-POSITION	USN-6
2	Decimal	USN.POST-TYPE	USN-11
4	Character	USN.RTE-INTSECT	USN-7
4	Character	USN.RTE-TRAV	USN-7

Length	Format	Data Element Name	Described Under
1	Decimal	USN.SHAPE	USN-32
1	Character	USN.SIDE-OF-STREET	USN-23
1	Character	USN.SIGN-#	USN-1
3	Character	USN.SIGN-AGE	USN-35
1∅	Character	USN.SIGN-CODE-#	USN-27
1	Decimal	USN.SIGN-CONDITION	USN-18
1	Decimal	USN.SIGN-POSITION	USN-19
3	Decimal	USN.SGN:COST-1	USN-16
3	Decimal	USN.SGN:COST-2	USN-16
3	Decimal	USN.SGN:COST-3	USN-16
4	Decimal	USN.SGN:COST-4	USN-16
6	Date	USN.SGN:DATE	USN-3
2	Decimal	USN.SGN:DAY	USN-3
2	Decimal	USN.SGN:HOUR	USN-3
6	Date	USN.SGN:INS-DATE	USN-4
2	Decimal	USN.SGN:INS-DAY	USN-4
1	Character	USN.SGN:INS-INV-CODE	USN-4
2	Decimal	USN.SGN:INS-MONTH	USN-4
2	Decimal	USN.SGN:INS-YEAR	USN-4
2	Decimal	USN.SGN:MONTH	USN-3
3	Character	USN.SGN:SEQ-#	USN-1
6	Date	USN.SGN:UPDATE-DATE	USN-17
2	Decimal	USN.SGN:UPDATE-DAY	USN-17
2	Decimal	USN.SGN:UPDATE-MONTH	USN-17
2	Decimal	USN.SGN:UPDATE-YEAR	USN-17
4	Character	USN.SGN:UPDTE-CD	USN-2
2	Decimal	USN.SGN:YEAR	USN-3
2	Character	USN.SUPP-CODE	USN-27
1	Character	USN.TIE-BREAKER	USN-1
6	Character	USN.TIE-BRKR-&-ASMBLY-#	USN-1
1	Character	USN.UNIFORM	USN-2∅
3	Decimal	USN.VERT-DIM	USN-33
5	Character	USN.VISIB-OTHER-RTE-1	USN-22
5	Character	USN.VISIB-OTHER-RTE-2	USN-22
1	Decimal	USN.VISIBILITY	USN-13
4	Decimal	USN.X-COORD	USN-8
6	Decimal	USN.X-FEET	USN-15
4	Decimal	USN.Y-COORD	USN-8
6	Decimal	USN.Y-FEET	USN-15

Listing of Names in File Order

Length	Format	Data Element Name	Described Under
1	Character	USN.TIE-BREAKER	USN-1
5	Character	USN.ASMBLY-#	USN-1
6	Character	USN.TIE-BRKR-&-ASMBLY-#	USN-1
1	Character	USN.SIGN-#	USN-1
3	Character	USN.ASM:SEQ-#	USN-1
3	Character	USN.SGN:SEQ-#	USN-1
4	Character	USN.ASM:UPDTE-CD	USN-2
4	Character	USN.SGN:UPDTE-CD	USN-2
6	Date	USN.ASM:DATE	USN-3
2	Decimal	USN.ASM:DAY	USN-3
2	Decimal	USN.ASM:MONTH	USN-3
2	Decimal	USN.ASM:YEAR	USN-3
2	Decimal	USN.ASM:HOUR	USN-3
6	Date	USN.SGN:DATE	USN-3
2	Decimal	USN.SGN:DAY	USN-3
2	Decimal	USN.SGN:MONTH	USN-3
2	Decimal	USN.SGN:YEAR	USN-3
2	Decimal	USN.SGN:HOUR	USN-3
6	Date	USN.ASM:INS-DATE	USN-4
2	Decimal	USN.ASM:INS-DAY	USN-4
2	Decimal	USN.ASM:INS-MONTH	USN-4
2	Decimal	USN.ASM:INS-YEAR	USN-4
1	Character	USN.ASM:INS-INV-CODE	USN-4
6	Date	USN.SGN:INS-DATE	USN-4
2	Decimal	USN.SGN:INS-DAY	USN-4
2	Decimal	USN.SGN:INS-MONTH	USN-4
2	Decimal	USN.SGN:INS-YEAR	USN-4
1	Character	USN.SGN:INS-INV-CODE	USN-4
1	Decimal	USN.POST-CONDITION	USN-5
1	Decimal	USN.POST-POSITION	USN-6
14	Character	USN.ASMBLY-LOCATION	USN-7
4	Character	USN.RTE-TRAV	USN-7
1	Character	USN.DIR-TRAV	USN-7
4	Character	USN.RTE-INTSECT	USN-7
5	Character	USN.DIST-FROM-INTSECT	USN-7
4	Decimal	USN.X-COORD	USN-8
4	Decimal	USN.Y-COORD	USN-8
3	Character	USN.MAP-NUMBER	USN-9
1	Character	USN.ASMBLY-TYPE	USN-1Ø
2	Decimal	USN.POST-TYPE	USN-11
1	Decimal	USN.#-POSTS	USN-12
1	Decimal	USN.VISIBILITY	USN-13
2	Decimal	USN.CHNGE-MTNG-HT	USN-14
2	Decimal	USN.CHNGE-LAT-CLR	USN-14
6	Decimal	USN.X-FEET	USN-15
6	Decimal	USN.Y-FEET	USN-15

Length	Format	Data Element Name	Described Under
3	Decimal	USN.ASM:COST-1	USN-16
3	Decimal	USN.ASM:COST-2	USN-16
3	Decimal	USN.ASM:COST-3	USN-16
4	Decimal	USN.ASM:COST-4	USN-16
3	Decimal	USN.SGN:COST-1	USN-16
3	Decimal	USN.SGN:COST-2	USN-16
3	Decimal	USN.SGN:COST-3	USN-16
4	Decimal	USN.SGN:COST-4	USN-16
6	Date	USN.ASM:UPDATE-DATE	USN-17
2	Decimal	USN.ASM:UPDATE-MONTH	USN-17
2	Decimal	USN.ASM:UPDATE-DAY	USN-17
2	Decimal	USN.ASM:UPDATE-YEAR	USN-17
6	Date	USN.SGN:UPDATE-DATE	USN-17
2	Decimal	USN.SGN:UPDATE-MONTH	USN-17
2	Decimal	USN.SGN:UPDATE-DAY	USN-17
2	Decimal	USN.SGN:UPDATE-YEAR	USN-17
1	Decimal	USN.SIGN-CONDITION	USN-18
1	Decimal	USN.SIGN-POSITION	USN-19
1	Character	USN.UNIFORM	USN-2Ø
1	Character	USN.DIRECTION-FACING	USN-21
5	Character	USN.VISIB-OTHER-RTE-1	USN-22
5	Character	USN.VISIB-OTHER-RTE-2	USN-22
1	Character	USN.SIDE-OF-STREET	USN-23
1	Decimal	USN.FUNCT-CLASS	USN-24
1	Character	USN.FED-AID-SYS	USN-25
1	Character	USN.MAINT-RESPON	USN-26
1Ø	Character	USN.SIGN-CODE-#	USN-27
2	Character	USN.SUPP-CODE	USN-27
2	Decimal	USN.COLOR	USN-28
1	Decimal	USN.LETTER-TYPE	USN-29
1	Decimal	USN.MATERIAL	USN-3Ø
1	Decimal	USN.FACE	USN-31
1	Decimal	USN.SHAPE	USN-32
3	Decimal	USN.HORIZ-DIM	USN-33
3	Decimal	USN.VERT-DIM	USN-33
2	Decimal	USN.MTNG-HT	USN-34
2	Decimal	USN.LAT-CLR	USN-34
3	Character	USN.SIGN-AGE	USN-35
4	Character	USN.CUR-SIGN-AGE	USN-35

USN-1 - Key - The urban sign key format is:

Columns	Length	Contents
1	1	Assembly number tie-breaker
2-6	5	Assembly number
7	1	Sign number
8-1Ø	3	Sequence number

The following data element names are used to refer to the key:

Data Element Name	Record Type	Columns	Length
USN.TIE-BRKR-&-ASMBLY-#	Both	1-6	6
USN.TIE-BREAKER	Both	1	1
USN.ASMBLY-#	Both	2-6	5
USN.SIGN-#	Sign	7	1
USN.ASM:SEQ-#	Assembly	8-1∅	3
USN.SGN:SEQ-#	Sign	8-1∅	3

USN-2 - Update Code - USN.ASM:UPDTE-CD contains the assembly update code and USN.SGN:UPDTE-CD contains the sign update code. The assembly update codes are:

'AINV'	New inventory	'ARPL'	Replacement
'AINS'	New installation	'ARPR'	Repair
'AREM'	Removal	'AVIS'	Change in visibility
'AREL'	Relocate	'ACND'	Change in assembly condition

The sign update codes are:

'SINV'	New inventory	'SRPL'	Replacement
'SINS'	New installation	'SRPR'	Repair
'SREM'	Removal	'SCND'	Change in sign condition
'SRPS'	Reposition		

Example: USN.ASM:UPDTE-CD *EQ* 'ARPR'.

USN-3 - Date of Maintenance - These fields (one for the assembly and one for the sign) contain the date of the information. The data element names are:

Data Element Name	Example
USN.ASM:DATE	USN.ASM:DATE *GE* 4/8/76
USN.ASM:MONTH	USN.ASM:MONTH *EQ* 6
USN.ASM:DAY	
USN.ASM:YEAR	USN.ASM:YEAR *GE* 76
USN.ASM:HOUR	USN.ASM:HOUR *EQ* 24
USN.SGN:DATE	
USN.SGN:MONTH	
USN.SGN:DAY	
USN.SGN:YEAR	Note: Unknown hour is stored as ∅. Times between midnight and 1:∅∅ A.M. are stored as 24.
USN.SGN:HOUR	

USN-4 - Date of Installation - These fields (one for the assembly and one for the sign) contain the original date (the date of installation or inventory or (signs only) replacement). A code is stored along with the date to indicate which type of date is stored:

'S' Date of installation
'V' Date of inventory
'R' Date of replacement (signs only)

Use the following data element names:

USN.ASM:INS-DATE	USN.SGN:INS-DATE
USN.ASM:INS-MONTH	USN.SGN:INS-MONTH
USN.ASM:INS-DAY	USN.SGN:INS-DAY
USN.ASM:INS-YEAR	USN.SGN:INS-YEAR
USN.ASM:INS-INV-CODE	USN.SGN:INS-INV-CODE

USN-5 - Post condition - USN.POST-CONDITION contains one of these codes:

Ø Satisfactory
1 Inadequate - replace
2 Damaged - vandalism - replace
3 Damaged - accidental - replace
4 Damaged - natural causes - replace
5 Damaged - vandalism - repair
6 Damaged - accidental - repair
7 Damaged - natural causes - repair
8 ---
9 Assembly does not exist but one should be erected

Example: USN.POST-CONDITION *EQ* 1.

USN-6 - Post position - USN.POST-POSITION contains one of these codes:

Ø Satisfactory
1 Should be relocated

Example: USN.POST-POSITION *EQ* 1.

USN-7 - Assembly Location - The assembly location has the following format:

Columns	Length	Contents
1-4	4	Route traveled
5	1	Direction traveled
6-9	4	Route intersected
1Ø-14	5	Distance from intersection

Use the following data element names:

Data Element Name	Columns	Length
USN.ASMBLY-LOCATION	1-14	14
USN.RTE-TRAV	1-4	4
USN.DIR-TRAV	5	1
USN.RTE-INTSECT	6-9	4
USN.DIST-FROM-INTSECT	1Ø-14	5

Example: USN.RTE-TRAV *EQ* '2533'.

USN-8 - X- and Y-Coordinates - USN.X-COORD and USN.Y-COORD contain 4-digit coordinates calculated from the assembly location. Example: USN.X-COORD *EQ* 12ØØ.

USN-9 - Map Number - USN.MAP-NUMBER contains the 3-character map number. Example: USN.MAP-NUMBER *EQ* '144'.

USN-1Ø - Assembly Type - USN.ASMBLY-TYPE contains one of these codes:

' '	One sign	'J'	Junction
'A'	Advance turn	'R'	Regulatory
'C'	Confirming route	'S'	School
'D'	Direction	'T'	Turn
'G'	Guide	'W'	Warning
'I'	Information	'X'	Combination

Example: USN.ASMBLY-TYPE *EQ* 'W'.

USN-11 - Post Type - USN.POST-TYPE contains one of these codes:

Ø1	Steel channel or U	15	Structure column
Ø2	Steel round	16	Tree
Ø3	Steel H beam	17	Fixed object
Ø4	Wood 4x4	18	Overhead truss
Ø5	Wood 4x6	19	Overpass (bridge)
Ø6	Wood 6x6	2Ø	Overhead span wire
Ø7	Wood round	21	Overhead cantilever
Ø8	Light pole	22	On back of another sign
Ø9	Power pole	23	Street sign pole
1Ø	Steel round breakaway	24	Wood 4x6 breakaway
11	Steel H beam breakaway	25	Wood 8x8 breakaway
12	Wood round breakaway	26	Telspar
13	Wood 6x6 breakaway	27	Wood 6x8 non-breakaway
14	Wood 6x8 breakaway	28	Wood 8x8 non-breakaway

Example: USN.POST-TYPE *EQ* 16.

USN-12 - Number of Posts - USN.#-POSTS contains a count of the number of posts. Example: USN.#-POSTS *EQ* 3.

USN-13 - Visibility - USN.VISIBILITY contains one of the following codes:

Ø Easily seen
1 Obscured by vegetation - one-time
2 Obscured by vegetation - periodic
3 Obscured by fixed object
4 Obscured by movable object

Example: USN.VISIBILITY *EQ* 2.

USN-14 - Change in Height or Clearance - USN.CHNGE-MTNG-HT contains the change in mounting height. USN.CHNGE-LAT-CLR contains the change in lateral clearance. Both fields are usually zero but can range from -99 to 99. Example: USN.CHNGE-LAT-CLR *LE* -4.

USN-15 - X- and Y-Feet - USN.X-FEET and USN.Y-FEET are 6-digit numbers that are byproducts of the calculation of x- and y-coordinates. They are retained because of their usefulness in the LIST-SIGNS-BY-STREET program.

USN-16 - Cost Fields - Use the following names to reference the cost fields:

Data Element Name	Data Element Name	Length
USN.ASM:COST-1	USN.SGN:COST-1	3
USN.ASM:COST-2	USN.SGN:COST-2	3
USN.ASM:COST-3	USN.SGN:COST-3	3
USN.ASM:COST-4	USN.SGN:COST-4	4

Example: USN.ASM:COST-3 *GE* 5ØØ.

USN-17 - Date of Update - Use the following names:

Data Element Name - Assembly	Data Element Name - Sign
USN.ASM:UPDATE-DATE	USN.SGN:UPDATE-DATE
USN.ASM:UPDATE-MONTH	USN.SGN:UPDATE-MONTH
USN.ASM:UPDATE-DAY	USN.SGN:UPDATE-DAY
USN.ASM:UPDATE-YEAR	USN.SGN:UPDATE-YEAR

USN-18 - Sign Condition - USN.SIGN-CONDITION contains one of these codes:

Ø Satisfactory
1 Non-uniform - replace
2 Damaged - vandalism - replace
3 Damaged - accidental - replace
4 Damaged - natural causes - replace
5 Damaged - vandalism - repair
6 Damaged - accidental - repair
7 Damaged - natural causes - repair
8 Improper application - remove
9 Sign does not exist but one should be erected

Example: USN.SIGN-CONDITION *NE* Ø.

USN-19 - Sign Position - USN.SIGN-POSITION contains one of these codes:

Ø Satisfactory
1 Should be repositioned

Example: USN.SIGN-POSITION *EQ* 1.

USN-2Ø - Uniform - USN.UNIFORM is a computer-generated field that contains one of the following codes:

' ' Sign is uniform
'*' Sign is non-uniform

Example: USN.UNIFORM *EQ* '*'.

USN-21 - Direction of Facing - USN.DIRECTION-FACING contains one of these codes:

'A'	Sign seen by traffic in A direction
'B'	Sign seen by traffic in B direction
'E'	Sign seen by traffic in either direction
'N'	Sign not seen by traffic on route

Example: USN.DIRECTION-FACING *EQ* 'E'.

USN-22 - Visibility on Other Routes - These two fields are referenced by USN.VISIB-OTHER-RTE-1 and USN.VISIB-OTHER-RTE-2. In most cases these fields contain blanks. Otherwise, they contain a 4-character route number followed by a 1-character direction code ('A','B', or 'E'). Special codes that may be used are 'BIKE' (bicyclists) and 'PED' (pedestrians).

USN-23 - Side of Street - USN.SIDE-OF-STREET contains one of these codes:

'L'	Left
'R'	Right
'E'	End
'O'	Overhead
'x'	Code developed by municipality to identify overhead lanes

Example: USN.SIDE-OF-STREET *EQ* 'E'.

USN-24 - Functional Classification - USN.FUNCT-CLASS contains one of these codes:

Ø	Interstate	5	Rural principal arterial
1	Urban principal arterial	6	Rural minor arterial
2	Urban minor arterial	7	Rural major collector
3	Urban collector	8	Rural minor collector
4	Urban local streets	9	Rural local road

Example: USN.FUNCT-CLASS *EQ* 2.

USN-25 - Federal Aid System - USN.FED-AID-SYS contains one of these codes:

'I'	Interstate	'S'	Secondary
'P'	Primary	'U'	Urban
'E'	Primary urban extension	'L'	Local (non federal-aid)

Example: USN.FED-AID-SYS *EQ* 'E'.

USN-26 - Maintenance Responsibility - USN.MAINT-RESPON contains one of these codes:

'M'	Municipality
'S'	State
'C'	County

Example: USN.MAINT-RESPON *EQ* 'M'.

USN-27 - Sign Code Number - USN.SIGN-CODE-# contains the 1Ø-character sign code number. USN.SUPP-CODE contains the 2-character supplemental code. Example: USN.SIGN-CODE-# *EQ* 'RØ1-Ø1-Ø2'.

USN-28 - Sign Color - USN.COLOR contains one of these codes:

ØØ	Yellow and black	Ø5	Brown and white
Ø1	Red and white	Ø6	Brown and tan
Ø2	Black and white	Ø7	Blue and white
Ø3	Green and white	Ø8	Orange and black
Ø4	Red white and blue		

USN-29 - Letter Type - USN.LETTER-TYPE contains one of these codes:

1 Demountable
2 Direct applied
3 Screened
4 Routed (engraved)

Example: USN.LETTER-TYPE *EQ* 4.

USN-3Ø - Material - USN.MATERIAL contains one of these codes:

1	Sheet aluminum increment	5	Wood
2	Extruded aluminum	6	Steel
3	Sheet aluminum	7	Fiberglass
4	plywood		

Example: USN.MATERIAL *EQ* 4.

USN-31 - Face - USN.FACE contains one of these codes:

1 Paint
2 Enamel
3 Reflective sheeting - engineer grade
4 Reflective sheeting - high-intensity grade
5 Reflective buttons on non-reflective background

Example: USN.FACE *EQ* 3.

USN-32 - Shape - USN.SHAPE contains one of these codes:

Ø	Pennant	5	Triangle
1	Octagon	6	Circular
2	Square	7	Pentagon
3	Rectangular	8	Trapezoid
4	Diamond	9	Shield

Example: USN.SHAPE *EQ* 4.

USN-33 - Dimensions - USN.HORIZ-DIM and USN.VERT-DIM contain the horizontal and vertical sign dimensions in inches. Each can range from Ø to 999. Example: USN.HORIZ-DIM *EQ* 3Ø.

USN-34 - Height and Clearance - USN.MTNG-HT and USN.LAT-CLR contain the mounting height and lateral clearance. Each is a 2-digit number and is in feet. Example: USN.MTNG-HT *LT* 1Ø.

USN-35 - Sign Age - USN.SIGN-AGE contains the sign age field as stored in the record. USN.CUR-SIGN-AGE contains the computed sign age (the sign age as stored plus the time that has passed from the date of installation until the current date). USN.SIGN-AGE is a 3-character field and contains years in the first character and months in the last two characters. Special codes are:

'ØØØ'	New
'999'	Older than 9 years 11 months
'UNK'	Unknown

USN.CUR-SIGN-AGE is a 4-character field and contains years in the first two characters and months in the last two characters. Special codes are:

'9999'	Sign age stored as '999'
'UNK'	Sign age unknown

APPENDIX C

ABEND CODES

This appendix describes briefly some of the ABEND codes that can be encountered when using HIS.

User Abend Codes

User codes are not widely used within HIS. The ones that can be encountered are:

User Code	Description
1Ø	The PARM field of the EXEC statement is in error
2Ø	An abend was requested via a DUMP parameter in the PARM field of the EXEC statement
1ØØ	Usually caused by a trap to the PL/I ON ERROR condition (i.e., the IBM software discovered an error condition that was not handled by the program). A user-1ØØ abend is usually accompanied by an IBM error message or by a HIS error message that identifies the cause of the error.
4ØØØ	This code is issued by the IBM software under certain conditions. It usually results from a serious programming error such as overwriting control data because of an unset pointer variable.

System Abend Codes

The system abend codes are described in detail in the IBM publication VS1 System Codes. Some of the ABEND codes that are sometimes encountered when running HIS are:

System Code	Description
Ø13	A file could not be opened. An accompanying IBM error message identifies which file could not be opened.
Ø31	An I/O error was detected with an ISAM file. The DCBEXCD1 and DCBEXCD2 fields of the DCB indicate possible cause. These fields are located at the DCB address plus X'5Ø'. The fields are laid out as shown:

DCBEXCD1 - A(DCB) + X'5Ø'

x...	Direct read to key higher than last in file
..x.	Space not found
.... xx..	Uncorrectable input or output error
.... ..xx	Block could not be reached (input or update)

System Code	Description
	DCBEXCD2 - A(DCB) + X'51' x... .... (output) sequence check (keys not written in order) .x.. .... (output) duplicate key ...x .... Overflow record
Ø3B	An ISAM file could not be opened. This abend will result when you attempt to write data to an ISAM file without specifying DISP=OLD. It can also be caused by incorrect DCB specifications (such as a blocksize that is not a multiple of the record length).
ØCx	Program interruption. This is sometimes caused by a missing DD statement. Codes that may be encountered often are: ØC1 - Operation exception (invalid op-code in instruction) ØC2 - Priveleged operation exception ØC4 - Protection exception (address outside range of partition) ØC5 - Addressing exception (address outside range of machine) ØC6 - Specification exception (wrong boundary) ØC7 - Data exception ØC8 - Fixed overflow ØC9 - Fixed divide ØCA - Decimal overflow ØCB - Decimal divide
122	Job was canceled by operator. A dump was requested.
213	A file could not be opened. The IBM message IEC143I identifies which file could not be opened.
222	Job was canceled by operator. No dump was requested.
322	Job was canceled by system because of time limitations. Default time is TIME=6 (6 minutes). You can specify more by coding TIME=x on the EXEC statement, but first be sure that the problem is not a programming error.
2F3	The IBM operating system failed during the job's execution. Re-submit the run.
5Ø6	Core limitations - run in larger partition
6Ø6	Core limitations - run in larger partition
7Ø6	The program is currently not usable - the link editor marked the load module as not executable
722	Job cancelled because too many printed lines were generated. Rerun the job and specify OUTLIM=xxx on the EXEC statement (xxx is the maximum number of lines that can be printed - estimate at 6Ø lines per page). Default is 2ØØØØ when running under HIS.
8Ø6	The program is currently not usable - no load module could be found. This code may result from a mispelled program name on the command.

System Code	Description
8ØA	Core limitations - run in larger partition
CØ3	One or more files were not closed - This error is not detected until after the HIS request had been completed.
D37	A file has used all of its space. If you are updating one of the tables in HIS.TABLES or a member of another partitioned data set, the partitioned data set needs to be compressed.
E37	A file has used all of its space